

РОСЖЕЛДОР
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)
Тихорецкий техникум железнодорожного транспорта
(ТТЖТ – филиал РГУПС)

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по организации самостоятельной работы обучающихся по дисциплине
«Операционные системы и среды»
для студентов специальности
09.02.01 Компьютерные системы и комплексы
III КУРС

Тихорецк, 2024 г.

УТВЕРЖДАЮ

**Заместитель директора по УР
Н.Ю. Шитикова**

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 28BE503D4C0B7CA71E633A7117E37C49
Владелец: Шитикова Наталья Юрьевна
Действителен: с 14.07.2023 до 06.10.2024

Методические рекомендации по организации самостоятельной работы обучающимися по дисциплине «Операционные системы и среды» для специальности 09.02.01 Компьютерные системы и комплексы.

Организация-разработчик: Тихорецкий техникум железнодорожного транспорта – филиал Федерального государственного бюджетного образовательного учреждения высшего образования «Ростовский государственный университет путей сообщения» (ТТЖТ – филиал РГУПС)

Разработчики:

Украинский А.В., преподаватель ТТЖТ - филиала РГУПС

Рекомендованы цикловой комиссией № 4 специальностей 09.02.01, 11.02.06, 38.02.01

Протокол заседания № 1 от «02» сентября 2024 г.

СОДЕРЖАНИЕ

1. Пояснительная записка	4
2. Особенности курса и основные требования рабочей программы.....	6
3. Требования к организации самостоятельной работы студентов и правила пользования методическими рекомендациями	10
4. Виды самостоятельных работ и формы контроля	7
5. Характеристика задания	8
6. Тематический план самостоятельной работы	14
7. Задания для самостоятельной работы студентов.....	16
8. Заключение	16
9. Приложение	77
10. Список рекомендуемой литературы.....	79

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические рекомендации для организации самостоятельной работы по дисциплине «Операционные системы и среды» предназначены для обучающихся третьих курсов специальности 09.02.01 Компьютерные системы и комплексы.

В связи с введением в образовательный процесс нового Федерального государственного образовательного стандарта все более актуальной становится задача организации самостоятельной работы студентов. Самостоятельная работа определяется как индивидуальная или коллективная учебная деятельность, осуществляемая без непосредственного руководства преподавателя, но по его заданиям и под его контролем.

Самостоятельная работа студентов является одной из основных форм внеаудиторной работы при реализации учебных планов и программ. В материалах для самостоятельной работы студентов представлен курс поддержки и совершенствования общеобразовательных, коммуникативных, информационных компетенций, обеспечивающих практическое выполнение заданий (поиск, набор и обработка данных) и продуктивного плана.

Самостоятельная работа студентов проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений студентов;
- углубления и расширения теоретических знаний;
- развития познавательных способностей и активности студентов: самостоятельности, ответственности и организованности, творческой инициативы;
- формирования самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации.

В процессе выполнения самостоятельной работы студенты получают практические умения и навыки:

- умение оперировать данными на информационном рынке;

- умения работать с информацией (кодировать, представлять, измерять);

- умения обрабатывать информацию средствами информатики.

учебные умения:

- использовать различные информационные источники;

- расспрашивать, описывать, сравнивать, исследовать, анализировать оценивать;

- проводить самостоятельный поиск необходимой информации;

специальные учебные умения:

- осуществлять эффективный и быстрый поиск нужной информации;

- организовывать работу на компьютере;

- выбирать оптимальное программное обеспечение для работы с информацией;

- излагать информацию средствами информатики.

Самостоятельная работа может проходить в лекционном кабинете, во время внеклассных мероприятий, дома.

Количество часов, отведенное на самостоятельную работу, соответствует учебному плану.

2. ОСОБЕННОСТИ КУРСА И ОСНОВНЫЕ ТРЕБОВАНИЯ РАБОЧЕЙ ПРОГРАММЫ

1.1 Место дисциплины в структуре основной образовательной программы:

Учебная дисциплина «Операционные системы и среды» является обязательной частью общепрофессионального цикла основной образовательной программы в соответствии с ФГОС СПО по специальности 09.02.01 Компьютерные системы и комплексы. Особое значение дисциплина имеет при формировании и развитии ОК 01; ОК 07; ПК 2.2; ПК 2.3; ПК 2.4.

1.2. Цель и планируемые результаты освоения дисциплины:

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания

Код ПК, ОК	Умения	Знания
ОК 01 ОК 07 ПК 2.2 ПК 2.3 ПК 2.4	Уметь: – использовать средства операционных систем и сред для обеспечения работоспособности вычислительной техники; – работать в конкретной операционной системе; – работать со стандартными программами операционной системы; – поддерживать приложения различных операционных систем	Знать: – состав и принципы работы операционных систем и сред; – понятие, основные функции, типы операционных систем; – машинно-зависимые свойства операционных систем: обработку прерываний, обслуживание ввода-вывода, управление виртуальной памятью; – принципы построения операционных систем; – способы организации поддержки устройств, драйверы оборудования; – понятие, функции и способы использования программного интерфейса операционной системы, виды пользовательского интерфейса.

3. ТРЕБОВАНИЯ К ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ И ПРАВИЛА ПОЛЬЗОВАНИЯ МЕТОДИЧЕСКИМИ РЕКОМЕНДАЦИЯМИ

Самостоятельная деятельность учащихся – внеаудиторная работа, предполагающая самостоятельное извлечение информации, её обработка (анализ и синтез), решение лингвистических задач.

Следовательно, требования к организации внеаудиторной деятельности касаются процесса поиска информации, источников информации и полученных на аудиторных занятиях предметных знаний, умений, навыков:

1. Умение пользоваться электронными ресурсами: сайтами, книгами и учебными пособиями, справочниками и словарями.

2. Умение пользоваться письменными источниками, находить нужную информацию.

3. Уметь работать с полученной информацией: обрабатывать её, сокращать, конспектировать.

4. Учащиеся должны вести записи в лекционной тетради и уместно использовать записи при решении заданий самостоятельной работы.

5. Учащиеся должны выполнять требования к каждому заданию данных методических рекомендаций.

Структура методических рекомендаций создана таким образом, чтобы максимально облегчить работу студентам и преподавателям. Пользоваться данными указаниями несложно: необходимо внимательно прочесть требования к выполнению заданий, просмотреть списки рекомендуемой литературы.

4. ВИДЫ САМОСТОЯТЕЛЬНЫХ РАБОТ И ФОРМЫ КОНТРОЛЯ

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Содержание внеаудиторной самостоятельной определяется в соответствии с рекомендуемыми видами заданий согласно примерной и рабочей программ учебной дисциплины.

Виды заданий для самостоятельной работы:

На основании компетентного подхода к реализации профессиональных образовательных программ, видами заданий для внеаудиторной самостоятельной работы являются:

Для овладения знаниями: поиск информации в сети Интернета, проведение исследований, подготовка сообщений.

Для закрепления и систематизации знаний: применение различных специализированных программ.

Для формирования умений: обработка информации прикладными программами, проектирование и моделирование объектов.

Формы самостоятельной работы:

Поиск информации в различных источниках и ее практическая обработка.

Самостоятельная работа в виде выполнения заданий.

Составление информационных моделей объектов и их анализ.

Критерии оценки результатов внеаудиторной самостоятельной работы студентов:

- уровень освоения студентами учебного материала;
- умение студента использовать теоретические знания при выполнении практических задач;
- сформированность общеучебных умений;
- обоснованность и четкость изложения ответа;
- оформление материала в соответствии с требованиями.

Контроль выполненной самостоятельной работы осуществляется индивидуально, на уроке, при тестировании, на семинаре, при защите рефератов и проектов:

Контроль сообщений осуществляется на уроках.

Контроль выполнения рефератов осуществляется индивидуальной (или групповой) беседой по ключевым моментам работы, с последующей защитой реферата.

Проверка самостоятельных работ информационных моделей объектов проверяется индивидуально. Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

Контроль результатов внеаудиторной самостоятельной работы студентов может осуществляться в пределах времени, отведенного на обязательные учебные занятия по дисциплине и внеаудиторную самостоятельную работу студентов по дисциплине, может проходить в письменной, устной или смешанной форме.

5. ХАРАКТЕРИСТИКА ЗАДАНИЯ

1. Написание реферата – это более объемный, чем сообщение, вид самостоятельной работы студента, содержащий информацию, дополняющую и развивающую основную тему, изучаемую на аудиторных занятиях (*приложение*). Ведущее место занимают темы, представляющие профессиональный интерес, несущие элемент новизны. Реферативные материалы должны представлять письменную модель первичного документа – научной работы, монографии, статьи. Реферат может включать обзор нескольких источников и служить основой для доклада на определенную тему на семинарах, конференциях.

Регламент озвучивания реферата – 7-10 мин.

Затраты времени на подготовку материала зависят от трудности сбора информации, сложности материала по теме, индивидуальных особенностей студента и определяются преподавателем. Ориентировочное время на подготовку – 4 ч.

Порядок сдачи и защиты рефератов.

1. Реферат сдается на проверку преподавателю за 1-2 недели до зачетного занятия
2. При оценке реферата преподаватель учитывает
 - качество
 - степень самостоятельности студента и проявленную инициативу
 - связность, логичность и грамотность составления
 - оформление в соответствии с требованиями ГОСТ.
3. Защита тематического реферата может проводиться на выделенном одном занятии в рамках часов учебной дисциплины или конференции или по одному реферату при изучении соответствующей темы, либо по договоренности с преподавателем.
4. Защита реферата студентом предусматривает
 - доклад по реферату не более 5-7 минут
 - ответы на вопросы оппонента.

На защите запрещено чтение текста реферата.

Общая оценка за реферат выставляется с учетом оценок за работу, доклад, умение вести дискуссию и ответы на вопросы. Содержание и оформление разделов реферата.

Титульный лист является первой страницей реферата и заполняется по строго определенным правилам.

В верхнем поле указывается полное наименование учебного заведения.

В среднем поле дается заглавие реферата, которое проводится без слова «тема» и в кавычки не заключается.

Далее, ближе к правому краю титульного листа, указываются фамилия, инициалы студента, написавшего реферат, а также его курс и группа. Немного ниже или слева указываются название кафедры, фамилия и инициалы преподавателя - руководителя работы.

В нижнем поле указывается год написания реферата.

После титульного листа помещают оглавление, в котором приводятся все заголовки работы и указываются страницы, с которых они начинаются. Заголовки оглавления должны точно повторять заголовки в тексте. Сокращать их или давать в другой формулировке и последовательности нельзя.

Все заголовки начинаются с прописной буквы без точки на конце. Последнее слово каждого заголовка соединяют отточием (.....) с соответствующим ему номером страницы в правом столбце оглавления.

Заголовки одинаковых ступеней рубрикации необходимо располагать друг под другом. Заголовки каждой последующей ступени смещают на три - пять знаков вправо по отношению к заголовкам предыдущей ступени.

Введение. Здесь обычно обосновывается актуальность выбранной темы, цель и содержание реферата, указывается объект (предмет) рассмотрения, приводится характеристика источников для написания работы и краткий обзор имеющейся по данной теме литературы. Актуальность предполагает оценку своевременности и социальной значимости выбранной темы, обзор литературы по теме отражает знакомство автора реферата с имеющимися источниками, умение их систематизировать, критически рассматривать, выделять существенное, определять главное.

Основная часть. Содержание глав этой части должно точно соответствовать теме работы и полностью ее раскрывать. Эти главы должны показать умение исследователя сжато, логично и аргументировано излагать материал, обобщать, анализировать, делать логические выводы.

Заключительная часть. Предполагает последовательное, логически стройное изложение обобщенных выводов по рассматриваемой теме.

Библиографический список использованной литературы составляет одну из частей работы, отражающей самостоятельную творческую работу автора, позволяет судить о степени фундаментальности данного реферата.

В работах используются следующие способы построения библиографических списков: по алфавиту фамилий, авторов или заглавий; по тематике; по видам изданий; по характеру содержания; списки смешанного построения. Литература в списке указывается в алфавитном порядке (более распространенный вариант - фамилии авторов в алфавитном порядке), после указания фамилии и инициалов автора указывается название литературного источника, место издания (пишется сокращенно, например, Москва - М., Санкт - Петербург - СПб ит.д.), название издательства (например, Мир), год издания (например, 2022), можно указать страницы (например, с. 54-67). Страницы можно указывать прямо в тексте, после указания номера, под которым литературный источник находится в списке литературы (например, 7 (номер лит. источника), с. 67- 89). Номер литературного источника указывается после каждого нового отрывка текста из другого литературного источника.

В приложении помещают вспомогательные или дополнительные материалы, которые загромождают текст основной части работы (таблицы, карты, графики, неопубликованные документы, переписка и т.д.). Каждое приложение должно начинаться с нового листа (страницы) с указанием в правом верхнем углу слова " Приложение" и иметь тематический заголовок. При наличии в работе более одного приложения они нумеруются арабскими цифрами (без знака " № "), например, " Приложение 1". Нумерация страниц, на которых даются приложения, должна быть сквозной и продолжать общую нумерацию страниц основного текста. Связь основного текста с приложениями осуществляется через ссылки,

которые употребляются со словом " смотри " (оно обычно сокращается и заключается вместе с шифром в круглые скобки - (см. прил. 1)).

Критерии оценки реферата

- актуальность темы, 1 балл;
- соответствие содержания теме, 3 балла;
- глубина проработки материала, 3 балла;
- грамотность и полнота использования источников, 1 балл;
- соответствие оформления реферата требованиям, 2 балла;
- доклад, 5 баллов;
- умение вести дискуссию и ответы на вопросы, 5 баллов.

Максимальное количество баллов: 20.

19-20 баллов соответствует оценке «5»

15-18 баллов – «4»

10-14 баллов – «3»

менее 10 баллов – «2»

2. Создание материалов-презентаций – это вид самостоятельной работы студентов по созданию наглядных информационных пособий, выполненных с помощью мультимедийной компьютерной программы. Этот вид работы требует координации навыков студента по сбору, систематизации, переработке информации, оформления ее в виде подборки материалов, кратко отражающих основные вопросы изучаемой темы, в электронном виде. То есть создание материалов-презентаций расширяет методы и средства обработки и представления учебной информации, формирует у студентов навыки работы на компьютере.

Материалы-презентации готовятся студентом в виде слайдов с использованием программы для презентаций. В качестве материалов-презентаций могут быть представлены результаты любого вида внеаудиторной самостоятельной работы, по формату соответствующие режиму презентаций.

Затраты времени на создание презентаций зависят от степени трудности материала по теме, его объема, уровня сложности создания презентации, индивидуальных особенностей студента и определяются преподавателем.

Ориентировочное время на подготовку – 1 ч.

Критерии оценки

- соответствие содержания теме, 1 балл;
- правильная структурированность информации, 5 баллов;
- наличие логической связи изложенной информации, 5 балл;
- эстетичность оформления, его соответствие требованиям, 3 балла;
- работа представлена в срок, 1 балл.

Максимальное количество баллов: 15.

14-15 баллов соответствует оценке «5»

11-13 баллов – «4»

8-10 баллов – «3»

менее 8 баллов – «2»

3. Подготовка презентация и доклада.

Доклад – это сообщение по заданной теме, с целью внести знания из дополнительной литературы, систематизировать материал, проиллюстрировать примерами, развивать навыки самостоятельной работы с научной литературой, познавательный интерес к научному познанию.

Тема доклада должна быть согласована с преподавателем и соответствовать теме занятия.

Материалы при его подготовке, должны соответствовать научно-методическим требованиям учебного заведения и быть указаны в докладе.

Необходимо соблюдать регламент, оговоренный при получении задания.

Иллюстрации должны быть достаточными, но не чрезмерными.

Работа студента над докладом-презентацией включает отработку навыков ораторства и умения организовать и проводить диспут.

Студент в ходе работы по презентации доклада, отрабатывает умение ориентироваться в материале и отвечать на дополнительные вопросы слушателей.

Студент в ходе работы по презентации доклада, отрабатывает умение самостоятельно обобщить материал и сделать выводы в заключении.

Докладом также может стать презентация реферата студента, соответствующая теме занятия.

Студент обязан подготовить и выступить с докладом в строго отведенное время преподавателем, и в срок.

Необходимо помнить, что выступление состоит из трех частей: вступление, основная часть и заключение.

Вступление помогает обеспечить успех выступления по любой тематике. Вступление должно содержать:

- название презентации (доклада)
- сообщение основной идеи
- современную оценку предмета изложения
- краткое перечисление рассматриваемых вопросов
- живую интересную форму изложения
- акцентирование оригинальности подхода

Основная часть, в которой выступающий должен глубоко раскрыть суть затронутой темы, обычно строится по принципу отчета. Задача основной части - представить достаточно данных для того, чтобы слушатели и заинтересовались темой и захотели ознакомиться с материалами. При этом логическая структура теоретического блока не должна даваться без наглядных пособий, аудио-визуальных и визуальных материалов.

Заключение - это ясное четкое обобщение и краткие выводы, которых всегда ждут слушатели.

Примерный план публичного выступления

1. Приветствие

«Добрый день!»

«Уважаемый «(имя и отчество преподавателя)

«Уважаемые присутствующие!»

2. Представление (Ф.И., группа, и т.д.)

«Меня зовут...Я учащийся (-щаяся).группы, техникума №..., города....»

3. Цель выступления

«Цель моего выступления – дать новую информацию по теме.

4. Название темы

«Название темы»

5.Актуальность

«Актуальность и выбор темы определены следующими факторами: во-первых,..., во-вторых,...»

6. Кратко о поставленной цели и способах ее достижения

«Цель моего выступления – ... основные задачи и способы их решения: 1..., 2..., 3...»

получены новые знания следующего характера:...,

выдвинуты новые гипотезы и идеи:...,

определены новые проблемы (задачи)»

7. Благодарность за внимание

«Благодарю за проявленное внимание к моему выступлению»

8. Ответы на вопросы

«Спасибо (благодарю) за вопрос...

А) Мой ответ...

Б) У меня, к сожалению, нет ответа, т.к. рассмотрение данного вопроса не входило в задачи моего исследования.

9. Благодарность за интерес и вопросы по теме

«Благодарю за интерес и вопросы по подготовленной теме. Всего доброго»

Факторы, влияющие на успех выступления

До, во время и после выступления на конференции докладчику необходимо учесть существенные факторы, непосредственно связанные с формой выступления - это внешний вид и речь докладчика, используемый демонстрационный материал, а также формы ответов на вопросы в ходе выступления.

Речь

Громкость – доступная для восприятия слов отдаленными слушателями, но без крика и надрыва.

Произношение слов – внятное, четкое, уверенное, полное (без глотания окончаний), с правильным литературным ударением.

Темп – медленный – в значимых зонах информации, средний – в основном изложении, быстрый – во вспомогательной информации.

Интонация – дружественная, спокойная, убедительная, выразительная, без ироничных и оскорбительных оттенков.

Критерии оценки доклада

- актуальность темы, 1 балл;
- соответствие содержания теме, 1 балл;
- глубина проработки материала, 1 балл;
- грамотность и полнота использования источников, 1 балл;
- соответствие оформления доклада требованиям, 1 балл.
- умение вести дискуссию и ответы на вопросы, 5 баллов.

Максимальное количество баллов: 10.

9-10 баллов соответствует оценке «5»

7-8 баллов – «4»

5-7 баллов – «3»

менее 5 баллов – «2»

4. Решение задач

Решение задачи можно условно разбить на четыре этапа и в соответствии с данными этапами установить критерии оценки:

1. Ознакомиться с условием задачи (анализ условия задачи и его наглядная интерпретация схемой или чертежом), 0,5 балл.
2. Составить план решения задачи (составление уравнений, связывающих физические величины, которые характеризуют рассматриваемое явление с количественной стороны), 2 балла;
3. Осуществить решение (совместное решение полученных уравнений относительно той или иной величины, считающейся в данной задаче неизвестной), 2 балла;
4. Проверка правильности решения задачи (анализ полученного результата и числовой расчет), 0,5 балла.

Максимальное количество баллов: 5.

Оценка выставляется по количеству набранных баллов. - Проведение контрольных и проверочных работ, включающих в себя задания самостоятельных;

- устный опрос;

- проверка выполненного материала

5. Выполнение тестовых заданий

Тесты и задания сориентированы на проверку выполнения обязательных требований к уровню общеобразовательной подготовки по физике.

Система заданий возрастающей степени трудности и специфической формы позволяет качественно оценить структуру и определить уровень знаний.

Тест состоит из следующих частей:

1. Одинаковая инструкция для всех испытуемых, которая должна быть настолько проста и понятна, насколько это возможно.

2. Сами тестовые задания: а) задания открытой формы. Инструкцией к заданиям данного типа является одно слово «дополните». За правильный ответ студент получает один балл. б) задания закрытой формы – вопрос с вариантами ответов, один или несколько из которых правильные. Неправильные ответы должны быть такими, чтобы каждый из них мог привлечь внимание. Инструкцией к этому типу заданий является: «выберите один (несколько) правильных ответов». За правильный ответ студент получает один балл. в) задания на восстановление соответствия. Инструкцией является: «установите соответствие». Число баллов оценивается отдельно, причём число баллов равно числу правильно установленных соответствий. Студент, допустивший хотя бы одну ошибку, получает 0 баллов. г) задания на установление правильной последовательности. Инструкцией является: «установите правильную последовательность». Если ранги в расставлены правильно – студент получает один балл, если допущена хотя бы одна ошибка – ноль баллов.

3. Одинаковые правила оценки ответов в рамках принятой формы. Оценка «удовлетворительно» ставится, если студент набирает не менее 50% баллов и до 75%. Оценка «хорошо» - 76 – 90% заданий. Оценка «отлично» - 90% и выше.

Структура теста удовлетворяет следующим основным требованиям:

- о задания каждого типа располагаются в одном месте и в порядке возрастания сложности;

- о задания формулируются в логической форме высказывания;
- о основной текст задания содержит не более 7 – 8 слов;
- о задания должны выполняться быстро, не более, чем за 1-2 минуты.

При организации тематического контроля знаний необходимо учитывать, что:

- о тематическая тестовая контрольная работа обычно рассчитана на 30-45 минут;
- о имеет 4 варианта;
- о каждый вариант имеет 15 заданий, правильные ответы на которые предполагают усвоение учебного материала, содержание заданий должно включать все основные понятия, законы и явления, необходимые для усвоения.

6. ТЕМАТИЧЕСКИЙ ПЛАН САМОСТОЯТЕЛЬНОЙ РАБОТЫ

№ п.п	Тема	Объём часов с.р.	Задание	Деятельность студента	Форма контроля
1	Общие сведения об операционных системах	1 час	Ср № 1	Поисково-исполнительский и творческий	Отчет о проведении работы
2	Основная классификация операционных систем	1 час	Ср № 2	Репродуктивно-подражающий	Устный опрос, проверка материала
3	Администрирование операционных систем	1 час	Ср № 3	Активизация познавательной деятельности	Устный опрос, проверка материала
4	Работа в оболочке командной строки. PowerShell	1 час	Ср № 4	Репродуктивно-подражающий	Устный опрос, проверка материала
5	Работа в оболочке командной строки. CMD	1 час	Ср № 5	Поисково-исполнительский и творческий	Отчет о проведении работы
6	Файловая система	1 час	Ср № 6	Репродуктивно-подражающий	Устный опрос, проверка материала
7	Типы файлов	1 час	Ср № 7	Активизация познавательной деятельности	Устный опрос, проверка материала
8	Операции при работе с каталогами	1 час	Ср № 8	Репродуктивно-подражающий	Устный опрос, проверка материала
9	Операции при работе с файлами	1 час	Ср № 9	Поисково-исполнительский и творческий	Отчет о проведении работы
10	Установка и предварительная настройка ОС	1 час	Ср № 10	Репродуктивно-подражающий	Устный опрос, проверка материала
11	Реестр и конфигурационные файлы	1 час	Ср № 11	Активизация познавательной деятельности	Устный опрос, проверка материала
12	Устройство мобильных операционных систем	1 час	Ср № 12	Репродуктивно-подражающий	Устный опрос, проверка материала
13	Модели операционных систем	1 час	Ср № 13	Репродуктивно-подражающий	Устный опрос, проверка материала
14	Процессы	1 час	Ср № 14	Поисково-исполнительский и творческий	Отчет о проведении работы
15	Потоки	1 час	Ср № 15	Репродуктивно-подражающий	Устный опрос, проверка материала

16	Управление процессами в Linux	1 час	Ср № 16	Активизация познавательной деятельности	Устный опрос, проверка материала
17	Пользовательские скрипты в Unix	1 час	Ср № 17	Поисково-исполнительский и творческий	Отчет о проведении работы
18	Управление памятью	2 часа	Ср № 18	Поисково-исполнительский и творческий	Отчет о проведении работы
19	Системные вызовы	2 часа	Ср № 19	Репродуктивно-подражающий	Устный опрос, проверка материала
20	Выгружаемые и невыгружаемые ресурсы	2 часа	Ср № 20	Активизация познавательной деятельности	Устный опрос, проверка материала
21	Сегментация памяти	2 часа	Ср № 21	Репродуктивно-подражающий	Устный опрос, проверка материала
22	Основные понятия безопасности, классификация угроз	2 часа	Ср № 22	Поисково-исполнительский и творческий	Отчет о проведении работы
23	Технологии безопасности, механизмы защиты	2 часа	Ср № 23	Репродуктивно-подражающий	Устный опрос, проверка материала
24	Резервное копирование и восстановление данных в Windows	2 часа	Ср № 24	Активизация познавательной деятельности	Устный опрос, проверка материала
25	Резервное копирование и восстановление данных в Unix	2 часа	Ср № 25	Репродуктивно-подражающий	Устный опрос, проверка материала
26	Сетевая модель OSI	2 часа	Ср № 26	Поисково-исполнительский и творческий	Отчет о проведении работы
27	Основные протоколы передачи данных	2 часа	Ср № 27	Репродуктивно-подражающий	Устный опрос, проверка материала
28	Стеки протокола FTP	2 часа	Ср № 28	Активизация познавательной деятельности	Устный опрос, проверка материала
29	Стеки протокола SSH	2 часа	Ср № 29	Репродуктивно-подражающий	Устный опрос, проверка материала
30	Серверные дистрибутивы операционных систем	2 часа	Ср № 30	Поисково-исполнительский и творческий	Отчет о проведении работы
31	Проводной и беспроводной доступ к сети: устройства и кабели	2 часа	Ср № 31	Репродуктивно-подражающий	Устный опрос, проверка материала
32	Адресация в сети. Провайдеры. Понятие хостинга.	2 часа	Ср № 32	Поисково-исполнительский и творческий	Отчет о проведении работы

7. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная работа № 1 (1 час)

Тема: Общие сведения об операционных системах.

Цель работы: Понять роль и функции операционных систем, а также их принципы работы.

Краткие сведения из теории

Операционная система (ОС) — это программное обеспечение, которое управляет аппаратными ресурсами компьютера и предоставляет среду для выполнения приложений. Операционная система служит посредником между пользователем и аппаратурой, обеспечивая удобный интерфейс для работы с системой.

Основные функции операционных систем:

1. Управление процессами:
 - ОС отвечает за создание, планирование и завершение процессов. Она управляет временем выполнения и ресурсами, необходимыми для выполнения процессов.
2. Управление памятью:
 - Операционная система управляет основным и внешним хранилищем, обеспечивая эффективное распределение и использование памяти.
3. Управление файлами:
 - ОС организует хранение и доступ к данным, предоставляя файловую систему для работы с файлами и их атрибутами.
4. Управление устройствами:
 - Операционная система управляет внешними устройствами (принтеры, диски, сетевые карты и др.) с помощью драйверов.
5. Обеспечение безопасности и защиты:
 - ОС реализует механизмы контроля доступа к ресурсам, защищая данные клиентов и обеспечивая безопасное выполнение программ.

Типы операционных систем:

1. Однопользовательские и многопользовательские:
 - Однопользовательские ОС предназначены для работы одного пользователя, тогда как многопользовательские поддерживают одновременный доступ нескольких пользователей.
2. Многозадачные и однозадачные:
 - Многозадачные ОС могут выполнять несколько процессов одновременно, тогда как однозадачные — только один.
3. Системы реального времени:
 - Эти ОС обеспечивают выполнение задач в строгие временные рамки, что критично для систем, таких как авиация или медицинские приборы.
4. Различные платформы:
 - Примеры операционных систем включают Windows, macOS, Linux, Android и iOS, каждая из которых предназначена для конкретных типов устройств и задач.

Задания для выполнения студентами

1. Исследование различных операционных систем:
 - Выберите две операционные системы (например, Windows и Linux) и подготовьте сравнительный анализ их функций, интерфейса и архитектуры.
2. Создание схемы:
 - Нарисуйте схему архитектуры операционной системы и обозначьте основные её компоненты (ядро, оболочка, файловая система, управление процессами и т.д.).

3. Практическое задание по командной строке:
 - На примере выбранной вами ОС выполните минимум пять команд в терминале или командной строке, которые связаны с управлением файлами и процессами. Запишите результаты.
4. Системы безопасности:
 - Подготовьте краткий обзор механизмов безопасности в выбранной вами операционной системе (например, Windows Defender на Windows или SELinux на Linux).
5. Управление памятью:
 - Исследуйте и опишите методы управления памятью в вашей операционной системе, включая виртуальную память и кэширование.

Контрольные вопросы по теме

1. Что такое операционная система и какие её основные функции?
2. Какие типы операционных систем существуют? Приведите примеры для каждого типа.
3. Как операционная система управляет процессами и ресурсами?
4. Что такое файловая система и какова её роль в операционной системе?
5. Какие механизмы обеспечивают безопасность и защиту данных в операционной системе?

Самостоятельная работа № 2 (1 час)

Тема: Основная классификация операционных систем.

Цель работы: Понять основные типы операционных систем и их назначение.

Краткие сведения из теории

Операционные системы (ОС) можно классифицировать по различным критериям, в зависимости от их назначения, архитектуры и функциональности. Вот основные категории:

1. По количеству пользователей:
 - Однопользовательские: предназначены для работы одного пользователя (например, Windows, macOS).
 - Многопользовательские: поддерживают одновременную работу нескольких пользователей (например, Unix, Linux).
2. По количеству задач:
 - Однозадачные: выполняют одну задачу в данный момент времени (например, MS-DOS).
 - Многозадачные: могут выполнять несколько задач одновременно, используя многопоточность или временную мультиплексацию (например, Linux, Windows NT).
3. По способу выполнения задач:
 - Пакетные: задачи выполняются в пакетах без взаимодействия с пользователем (например, старые системы IBM).
 - Интерактивные: позволяют пользователям взаимодействовать с системой в реальном времени (например, современные ОС на базе Unix).
4. По типу архитектуры:
 - Ядровые (Monolithic): все основные функции находятся в одном ядре (например, Linux).
 - Микроядерные (Microkernel): только базовые функции находятся в ядре, остальные выполняются в пользовательском пространстве (например, Minix, QNX).
5. По типу платформы:
 - Desktopные: ОС для стационарных и мобильных компьютеров (например, Windows, macOS).
 - Серверные: ОС, предназначенные для серверных решений (например, Windows Server, Red Hat Enterprise Linux).
 - Мобильные: ОС для мобильных устройств (например, Android, iOS).
 - Встраиваемые: установлены на специализированных устройствах (например, системы на базе FreeRTOS, VxWorks).
6. По системе реального времени:
 - ОС реального времени (RTOS): предназначены для выполнения задач в строгие временные рамки (например, QNX, RTEMS).

Задания для выполнения студентами

1. Исследование классов ОС:
 - Определите и подготовьте краткое описание каждой категории операционных систем, приведенной в теоретической справке.
2. Сравнительный анализ:
 - Сравните две операционные системы разных типов (например, Windows и Linux) по следующим критериям: управление процессами, управление памятью, интерфейс пользователя и поддержка многопользовательского режима.
3. Создание презентации:

- Разработайте презентацию о современных мобильных операционных системах, описывая их особенности, архитектуру и распространенные приложения.
- 4. Исследование реального времени:
 - Выберите одну операционную систему реального времени и изучите её применение в промышленности (например, в automotive или медицине).
- 5. Практическое задание:
 - Установите и протестируйте две разные операционные системы (например, Windows и Ubuntu Linux) на виртуальной машине. Подготовьте отчет о своем опыте.

Контрольные вопросы по теме

1. Каковы основные критерии классификации операционных систем?
2. В чем разница между однопользовательскими и многопользовательскими операционными системами?
3. Что такое микроядерная архитектура, и какие преимущества она предоставляет?
4. Как отличить пакетные операционные системы от интерактивных?
5. Каковы особенности операционных систем реального времени, и где они применяются?

Самостоятельная работа № 3 (1 час)

Тема: Администрирование операционных систем.

Цель работы: Понять задачи, стоящие перед системными администраторами, и развить практические навыки в управлении операционными системами.

Краткие сведения из теории

Администрирование операционных систем — это процесс управления и поддержки компьютерных систем и сетей, включая настройку, эксплуатацию и обслуживание программного обеспечения и аппаратуры. Администраторы операционных систем (или системные администраторы) играют ключевую роль в обеспечении функционирования компьютерных исследований и сетевых инфраструктур.

Основные задачи администрирования:

1. Установка и настройка ОС:
 - Процесс включает выбор подходящей версии ОС, её установку, настройку параметров и оптимизацию под конкретные задачи.
2. Управление пользователями и правами доступа:
 - Создание учетных записей, назначение прав и ролей, мониторинг активности пользователей для защиты данных и ресурсов системы.
3. Мониторинг системы:
 - Отслеживание состояния системы, производительности и загрузки ресурсов (ЦП, память, диск) с использованием специальных инструментов.
4. Резервное копирование и восстановление данных:
 - Регулярное создание резервных копий данных и баз данных для предотвращения потерь информации в случае сбоев или атак.
5. Обновление и патчинг программного обеспечения:
 - Установка обновлений безопасности и исправлений для программного обеспечения, чтобы обеспечить защиту от уязвимостей.
6. Обеспечение безопасности:
 - Реализация мер по защите от вирусов и атак, настройка брандмауэров и антивирусов, управление шифрованием данных.
7. Управление сетевыми ресурсами:
 - Настройка сетевых устройств, управление IP-адресами, конфигурация маршрутизаторов и коммутаторов.
8. Устранение неполадок:
 - Диагностика и решение проблем, возникающих в работе системы, включая анализ логов и применение методов восстановления.

Задания для выполнения студентами

1. Исследование задач администрирования:
 - Подготовьте отчет о каждой из задач администрирования. Опишите важность каждой задачи и инструменты, используемые для их выполнения.
2. Практическое задание по созданию пользователей:
 - На выбранной операционной системе (Windows/Linux) создайте нескольких пользователей с разными правами доступа и подготовьте инструкцию по этому процессу.
3. Мониторинг системы:
 - Установите и настройте одно из средств мониторинга производительности (например, htop для Linux или Task Manager для Windows) и проанализируйте, какие ресурсы загружены.
4. Резервное копирование данных:

- Изучите методы резервного копирования данных в вашей операционной системе и создайте тестовую резервную копию важных файлов. Подготовьте отчет об этом процессе.
5. Обновление системы:
- Установите последние обновления для вашей ОС и обеспечьте документацию по проведенным изменениям и улучшениям.

Контрольные вопросы по теме

1. Какие основные задачи выполняет системный администратор?
2. Как осуществляется мониторинг производительности системы?
3. Почему важна задача резервного копирования данных, и какие методы для этого существуют?
4. Какие меры необходимо предпринять для обеспечения безопасности операционной системы?
5. Какую роль играет управление пользователями в администрировании операционных систем?

Самостоятельная работа № 4 (1 час)

Тема: Работа в оболочке командной строки. PowerShell.

Цель работы: Получить практические знания о PowerShell и научиться эффективно использовать оболочку командной строки для администрирования и автоматизации задач. ,

Краткие сведения из теории

Оболочка командной строки — это интерфейс, который позволяет пользователям взаимодействовать с операционной системой через текстовые команды. PowerShell является мощной оболочкой командной строки, разработанной компанией Microsoft, и предоставляет среду для автоматизации и администрирования задач в системах Windows и других операционных системах.

Основные характеристики PowerShell:

1. Объектно-ориентированный: В отличие от традиционных оболочек, которые работают с текстом, PowerShell управляет объектами, что упрощает обработку данных.
2. Команды (cmdlets): PowerShell использует специальные команды, называемые cmdlets, которые выполняют конкретные функции (например, Get-Process, Set-Content).
3. Пайплайны: Возможность передачи выходных данных одной команды в качестве входных данных другой команды, что позволяет строить сложные сценарии.
4. Скриптовый язык: PowerShell поддерживает создание скриптов для автоматизации различных задач, что повышает эффективность работы администратора.
5. Интеграция с .NET: PowerShell построен на .NET Framework, что позволяет использовать возможности библиотеки для выполнения сложных операций.
6. Удаленное администрирование: Позволяет выполнять команды на удаленных системах через сеть.

Задания для выполнения студентами

1. Основы команд PowerShell:
 - Ознакомьтесь с основными командами PowerShell (cmdlets) и напишите их описание. Примеры команд: Get-Help, Get-Process, Set-Location.
2. Создание и выполнение простого скрипта:
 - Напишите скрипт PowerShell, который выводит информацию о текущем времени, дате и файловой системе, и сохраните его в файл с расширением .ps1. Запустите скрипт и подготовьте отчет о его выполнении.
3. Использование пайплайнов:
 - Создайте команду с использованием пайплайнов, которая выводит список процессов, сортирует их по использованию памяти и отображает только 5 верхних.
4. Работа с файлами:
 - Напишите PowerShell-команды для создания новой папки, копирования в неё текстового файла и вывода содержимого этого файла на экран.
5. Удаленное выполнение команд:
 - Изучите, как можно использовать PowerShell для удаленного администрирования. Проведите тестовое подключение к удаленному компьютеру с использованием команды Enter-PSSession и выполните простую команду, например Get-Service.

Контрольные вопросы по теме

1. Что такое PowerShell и каковы его основные преимущества по сравнению с традиционными оболочками?
2. Как определить, какие cmdlets доступны в текущей среде PowerShell?

3. В чем разница между объектно-ориентированным подходом PowerShell и текстовыми оболочками?
4. Как выглядит базовая структура скрипта PowerShell?
5. Какие меры предосторожности следует соблюдать при выполнении удаленных команд с использованием PowerShell?

Самостоятельная работа № 5 (1 час)

Тема: Работа в оболочке командной строки. CMD.

Цель работы: Ознакомиться с основами работы в CMD, изучить команды и скрипты для администрирования и автоматизации задач в операционной системе Windows.

Краткие сведения из теории

CMD (Command Prompt, командная строка) — это стандартная оболочка командной строки в операционных системах Windows. Она предоставляет пользователям возможность выполнять команды для управления файловой системой, запуска программ, а также для выполнения различных системных задач.

Основные характеристики CMD:

1. Текстовый интерфейс: Пользователи вводят команды в текстовом формате, а система отвечает текстом.
2. Команды и синтаксис: CMD использует команды, такие как `dir`, `cd`, `copy`, и поддерживает параметры для изменения поведения команд.
3. Пакетная обработка: Система позволяет создавать скрипты (файлы с расширением `.bat`) для автоматизации выполнения последовательности команд.
4. Системное администрирование: CMD может использоваться для управления сетевыми настройками, управления пользователями и системными службами.
5. Диагностика и устранение неполадок: CMD предоставляет утилиты, такие как `ping`, `ipconfig`, и `chkdsk`, для диагностики и устранения проблем.

Задания для выполнения студентами

1. Ознакомление с командами:
 - Изучите основные команды CMD и подготовьте их краткое описание, включая примеры использования. Примеры команд: `dir`, `cd`, `copy`, `del`.
2. Работа с файлами:
 - Используйте CMD для создания новой папки и текстового файла внутри неё. Напишите команды, которые вы использовали, и результаты выполнения.
3. Скриптовое программирование:
 - Создайте простой пакетный файл (файл с расширением `.bat`), который выполняет последовательность команд: создает папку, копирует в неё файл и выводит список файлов в папке.
4. Проверка сетевых подключений:
 - Выполните команду `ping` для проверки доступности любого веб-сайта (например, `ping yandex.ru`) и проанализируйте результаты.
5. Диагностика системы:
 - Используйте команду `ipconfig` для получения информации о сетевых настройках вашего компьютера. Подготовьте отчет о выводе этой команды.

Контрольные вопросы по теме

1. Какие основные функции выполняет CMD в операционной системе Windows?
2. Каковы основные команды для навигации по файловой системе в CMD?
3. В чем заключается отличие между командой `copy` и командой `move`?
4. Как создать и запустить пакетный файл с помощью CMD?
5. Какие команды можно использовать для диагностики сетевых подключений и выяснения текущих настроек сети?

Самостоятельная работа № 6 (1 час)

Тема: Файловая система.

Цель работы: Глубже понять концепцию файловых систем и их важность для работы операционных систем.

Краткие сведения из теории

Файловая система (ФС) — это совокупность методов и структур данных, используемых операционной системой для управления файлами на носителях информации. Она позволяет пользователям и программам сохранять и извлекать данные, организуя их в логически понятном и доступном виде.

1. Цели и задачи файловой системы:

- Управление данными: Обеспечение хранения, организации и доступа к данным на носителе.
- Поддержка последовательности операций: Запись, чтение, удаление и модификация файлов.
- Безопасность и контроль доступа: Предоставление механизмов для ограничения доступа к данным.
- Забота об целостности данных: Защита от повреждений и потери информации.

2. Виды файловых систем:

- FAT (File Allocation Table): Используется в большинстве версий Windows и на USB-носителях.
- NTFS (New Technology File System): Основная файловая система для Windows, поддерживает большие объемы данных и расширенные функции безопасности.
- EXT (Extended File System): Широко используемая файловая система в Linux, с версиями EXT2, EXT3 и EXT4.
- HFS+ (Hierarchical File System Plus): Используется на системах macOS.
- APFS (Apple File System): Современная файловая система для устройств Apple, оптимизированная для SSD.

3. Физическая организация файловой системы:

- Блоки: Файлы хранятся в виде непрерывных блоков на физическом носителе. Каждый файл занимает определенное количество блоков.
- Файловые структуры: Включают в себя таблицы, списки и деревья, которые используются для управления адресацией и местоположением файлов.
- Разделы: Носитель может быть разделен на несколько логических единиц, каждая из которых может содержать свою файловую систему.

4. Структура файловой системы:

- Директорий: Структура, организующая файлы в иерархическом порядке.
- Метаданные: Информация о файлах, включая имя, размер, время создания и модификации, права доступа.
- Индексы: Используются для быстрого поиска и доступа к файлам.

Задания для выполнения студентами

1. Изучение различных файловых систем:

- Выберите две файловые системы (например, NTFS и EXT4) и сравните их по функциям, производительности и области применения. Подготовьте сравнительную таблицу.

2. Анализ физической организации файловой системы:

- Изучите, как файлы хранятся на примере вашей ОС (Windows или Linux). Опишите, как создаются и хранятся директории и файлы.

3. Создание и управление файлами:

- На практике создайте несколько файлов и директорий в вашей операционной системе. Измените их атрибуты и удалите, проконтролировав изменения на уровне файловой системы.
4. Работа с метаданными:
 - С использованием инструментов командной строки, таких как `ls -l` или `dir`, получите информацию о метаданных файлов в вашей системе. Запишите результаты и проанализируйте их.
 5. Исследование целостности данных:
 - Изучите механизмы восстановления данных для вашей файловой системы (например, Shadow Copies в NTFS или Journal в EXT3/EXT4). Подготовьте краткий отчет о работе этих механизмов.

Контрольные вопросы по теме

1. Каковы основные задачи файловой системы?
2. Назовите и опишите основные виды файловых систем.
3. Как осуществляется физическая организация файловой системы?
4. Что такое метаданные и какую роль они играют в файловых системах?
5. Как можно обеспечить целостность данных в файловой системе?

Самостоятельная работа № 7 (1 час)

Тема: Типы файлов.

Цель работы: Изучение типов файлов и их роли в операционных системах.

Краткие сведения из теории

Файлы — это структурированные наборы данных, которые хранятся на носителях информации и используются операционными системами для организации и управления данными. Каждый файл имеет свое имя, тип и атрибуты, определяющие его поведение и доступность.

1. Типы файлов:

- Текстовые файлы: Содержат данные в текстовом формате (например, .txt, .csv). Читаемы человеком и часто используются для хранения конфигурации, логов и документации.
- Бинарные файлы: Содержат данные в удобном для компьютера формате (например, .exe, .bin). Не могут быть прочитаны напрямую без специального программного обеспечения.
- Исполняемые файлы: Содержат код, который может быть выполнен операционной системой (например, .exe для Windows, .app для macOS).
- Мультимедийные файлы: Включают изображения, видео и аудио (например, .jpg, .mp4, .mp3). Все эти файлы требуют специального ПО для воспроизведения или обработки.
- Архивные файлы: Используются для сжатия и хранения нескольких файлов в одном (например, .zip, .rar). Они позволяют экономить место и упрощают управление данными.
- Системные файлы: Необходимы для работы операционной системы. Включают драйверы и библиотеки (например, .dll для Windows).

2. Атрибуты файлов:

- Имя файла: Уникальное название, которое позволяет идентифицировать файл.
- Расширение: Обозначает тип файла и помогает операционной системе определить, как его открыть.
- Размер: Определяет объем данных, хранящихся в файле.
- Права доступа: Ограничения на чтение, запись и выполнение файла, которые определяют, кто может взаимодействовать с ним.

Задания для выполнения студентами

1. Классификация файлов:

- Соберите примеры различных типов файлов на вашем компьютере. Укажите их типы и расширения, составив таблицу.

2. Работа с текстовыми и бинарными файлами:

- Создайте текстовый файл, запишите в него несколько строк и сохраните его. Затем попробуйте открыть этот файл в текстовом редакторе и в шестнадцатеричном редакторе. Опишите различия в содержимом.

3. Анализ мультимедийных файлов:

- Выберите три мультимедийных файла (изображение, аудио, видео) и выясните, какие форматы и кодеки используются. Напишите краткое описание каждого файла.

4. Создание и распаковка архивов:

- Создайте архив с несколькими файлами, используя программу для сжатия (например, WinRAR, 7-Zip). Попробуйте распаковать архив и обсудите, как это помогло в управлении файлами.

5. Изучение системных файлов:

- Исследуйте папку системных файлов вашей операционной системы. Определите, какие файлы являются критически важными для работы системы. Напишите краткий отчет о их функциях.

Контрольные вопросы по теме

1. Какие основные типы файлов существуют, и каковы их отличия?
2. Что такое расширение файла и какую роль оно играет?
3. Какие атрибуты файлов вы знаете, и как они влияют на управляющие операции?
4. Каковы функции архивных файлов, и почему они полезны в управлении данными?
5. Какие типы файлов используются в операционных системах для системных нужд? Приведите примеры.

Самостоятельная работа № 8 (1 час)

Тема: Операции при работе с каталогами.

Цель работы: Освоить практические навыки работы с каталогами в операционных системах и углубить их понимание файловых систем.

Краткие сведения из теории

Каталоги (или директории) — это специальные файлы в файловой системе, которые служат для организации других файлов и подкаталогов. Они обеспечивают иерархическую структуру хранения данных, что облегчает навигацию и управление файлами.

Основные операции с каталогами:

1. **Создание:** Процесс создания нового каталога для хранения файлов и других подкаталогов. В большинстве операционных систем для этого используются команды (например, `mkdir` в UNIX/Linux или `md` в Windows).
2. **Удаление:** Процесс удаления каталога и всех его вложенных файлов и подкаталогов. Для выполнения этой операции используется команда (например, `rmdir` в UNIX/Linux или `rd` в Windows), но для удаления непустых каталогов может потребоваться использовать рекурсивное удаление.
3. **Рекурсивное удаление:** Удаление каталога вместе со всеми его вложенными файлами и подкаталогами. В UNIX/Linux это достигается с помощью ключа `-r` (например, `rm -r`).
4. **Переименование:** Изменение имени каталога. В большинстве систем это можно сделать с помощью команды (например, `mv` в UNIX/Linux и `ren` в Windows).
5. **Копирование:** Копирование каталога и его содержимого в новое местоположение. Эта операция может потребовать специальных параметров для рекурсивного копирования (например, `cp -r` в UNIX/Linux).

Задания для выполнения студентами

1. **Создание каталогов:**
 - Создайте на вашем компьютере несколько каталогов с различными именами. Включите подкаталоги в каждый из них. Запишите команды, которые вы использовали.
2. **Удаление каталогов:**
 - Удалите один из созданных каталогов. Попробуйте удалить заполненный каталог и прокомментируйте, что произошло. Используйте как простую, так и рекурсивную команду удаления.
3. **Переименование каталога:**
 - Измените имя одного из созданных каталогов, описанное ранее. Запишите использованную команду и результат.
4. **Копирование каталогов:**
 - Создайте копию одного из каталогов с его содержимым в другое место. Проверьте, что все файлы и подкаталоги успешно скопированы в новую директорию.
5. **Рекурсивные операции:**
 - Создайте структуру каталогов с несколькими уровнями вложенности. Затем выполните рекурсивное удаление и копирование, записывая команды и результаты для каждой операции.

Контрольные вопросы по теме

1. Какие команды вы используете для создания и удаления каталогов в вашей операционной системе?
2. В чем разница между простой и рекурсивной операцией удаления каталога?

3. Каковы команды для переименования и копирования каталогов в UNIX/Linux и Windows?
4. Почему важно понимать операции с каталогами при работе с файловыми системами?
5. Какие потенциальные риски могут возникнуть при выполнении рекурсивных операций над каталогами?

Самостоятельная работа № 9 (1 час)

Тема: Операции при работе с файлами.

Цель работы: Изучение операций с файлами.

Краткие сведения из теории

Файлы — это базовые единицы хранения данных в операционных системах. Они могут содержать различные виды информации, такие как текст, изображения, исполняемые программы и т.д. Операции с файлами являются важными для управления данными и их обработки.

Основные операции с файлами:

1. **Создание:** Создание нового файла в заданном каталоге. Обычно осуществляется с помощью команды (например, `touch` в UNIX/Linux или `echo >` в Windows).
2. **Удаление:** Удаление файла из файловой системы. Используются команды `rm` в UNIX/Linux или `del` в Windows.
3. **Переименование:** Изменение имени существующего файла. Команды для этого: `mv` в UNIX/Linux и `ren` в Windows.
4. **Копирование:** Создание копии файла в указанном месте. Команды: `cp` в UNIX/Linux и `copy` в Windows.
5. **Создание жесткой ссылки:** Создание нового файла, который ссылается на те же данные, что и оригинальный файл. В UNIX/Linux используется команда `ln`, в Windows — `mklink`.
6. **Вывод содержимого файла:** Отображение данных из файла. Для этого в UNIX/Linux подходят команды `cat`, `less`, `more`, а в Windows — `type`.
7. **Вывод содержимого файла с условиями:** Фильтрация вывода в соответствии с заданными условиями, например, с использованием команд `grep` в UNIX/Linux и `findstr` в Windows.

Задания для выполнения студентами

1. **Создание файлов:**
 - Создайте несколько текстовых файлов с различными именами. Зафиксируйте использованные команды.
2. **Удаление файлов:**
 - Удалите один из созданных файлов и запишите, какую команду вы использовали. Попробуйте удалить файл, который не существует, и прокомментируйте результат.
3. **Переименование файлов:**
 - Переименуйте один из созданных файлов. Запишите использованную команду и результаты.
4. **Копирование файлов:**
 - Скопируйте один из файлов в другую директорию. Убедитесь, что копия успешно создана.
5. **Создание жесткой ссылки:**
 - Создайте жесткую ссылку на один из ваших файлов. Проверьте, что обе ссылки (оригинальный файл и жесткая ссылка) ссылаются на одни и те же данные.
6. **Вывод содержимого файла:**
 - Выведите содержимое одного из ваших текстовых файлов, используя соответствующую команду.
7. **Вывод содержимого файла с условиями:**

- Используйте условие для фильтрации вывода содержимого файла (например, вывести только строки, содержащие заданное слово) с помощью команды grep (или findstr в Windows).

Контрольные вопросы по теме

1. Какие команды используются для создания и удаления файлов в вашей операционной системе?
2. Что такое жесткая ссылка и как она отличается от символической ссылки?
3. Каковы команды для копирования и переименования файлов в UNIX/Linux и Windows?
4. Какие средства вы можете использовать для отображения содержимого файла и фильтрации вывода?
5. Почему важно уметь выполнять основные операции с файлами при работе с операционными системами?

Самостоятельная работа № 10 (1 час)

Тема: Установка и предварительная настройка ОС.

Цель работы: Формирование практических навыков установки и настройки операционных систем, что является основой для дальнейшей работы в области информационных технологий.

Краткие сведения из теории

Установка операционной системы (ОС) — это многослойный процесс, включающий подготовку аппаратного обеспечения, установку самой системы и ее последующую конфигурацию. Правильная установка ОС является ключом к стабильной и безопасной работе устройства.

1. Подготовка к установке:
 - Системные требования: Убедитесь, что аппаратное обеспечение соответствует минимальным требованиям ОС.
 - Резервное копирование данных: Рекомендуется сделать резервную копию важных данных на внешние носители.
 - Выбор метода установки: Возможны установка с USB-накопителя, DVD-диска или через сеть (PXE).
2. Этапы установки:
 - Загрузка инсталлятора: Выбор загрузочного устройства в BIOS/UEFI.
 - Форматирование разделов: Создание и/или удаление разделов на жестком диске.
 - Копирование файлов: Системные файлы копируются на жесткий диск.
 - Настройка системы: Ввод настроек, таких как язык, временная зона, учетные записи пользователей и безопасности.
3. Предварительная настройка после установки:
 - Установка драйверов: После установки необходимо обновить драйвера для корректной работы аппаратного обеспечения.
 - Обновление системы: Установка всех доступных обновлений для повышения безопасности и стабильности системы.
 - Настройка сетевых подключений: Конфигурация Wi-Fi, Ethernet и других сетевых интерфейсов.
 - Персонализация системы: Настройка внешнего вида, рабочего стола и установленных приложений.

Задания для выполнения студентами

1. Подготовка оборудования:
 - Исследуйте системные требования для установки одной из популярных ОС (например, Windows, Linux). Составьте отчет о совместимости вашего оборудования.
2. Настройка BIOS/UEFI:
 - Загрузитесь в BIOS/UEFI вашего компьютера и опишите процесс изменения порядка загрузки устройств для установки ОС.
3. Установка ОС:
 - Выполните установку выбранной ОС на виртуальную машину (например, с использованием VirtualBox или VMware). Подробно опишите все шаги процесса установки.
4. Предварительная настройка:
 - После установки ОС выполните настройку следующего:
 - Установите драйвера для аппаратного обеспечения (если применимо).

- Выполните обновление системы и установку необходимых приложений (браузер, офисный пакет и т.д.).
 - Настройте учетную запись пользователя и пароли.
5. Исследование различных методов установки:
- Сравните два метода установки операционных систем (например, установка с USB-накопителя и установка через сеть). Опишите плюсы и минусы каждого метода.

Контрольные вопросы по теме

1. Какие этапы включает в себя установка операционной системы?
2. Каковы минимальные системные требования для установки Windows и Linux?
3. Для чего используется BIOS/UEFI при установке ОС?
4. Какие шаги необходимо выполнить для предварительной настройки ОС после установки?
5. Почему важно обновлять драйвера и саму операционную систему после установки?

Самостоятельная работа № 11 (1 час)

Тема: Реестр и конфигурационные файлы.

Цель работы: Углубить знания в области реестра и конфигурационных файлов, а также развить практические навыки работы с этими компонентами.

Краткие сведения из теории

Регистры и конфигурационные файлы — важные компоненты операционных систем, использующиеся для хранения настроек и параметров, необходимых для функционирования системы и приложений.

1. Реестр Windows:

- Реестр — это централизованное хранилище, где ОС Windows сохраняет данные конфигурации для системы и установленных приложений.
- Структура реестра состоит из ключей, подотрядов и значений. Основные компоненты корневых ключей включают:
 - HKEY_LOCAL_MACHINE: Конфигурация оборудования и программ для всех пользователей.
 - HKEY_CURRENT_USER: Настройки, специфичные для текущего пользователя.
 - HKEY_CLASSES_ROOT: Информация о зарегистрированных типах файлов и их связях с приложениями.

2. Конфигурационные файлы:

- Используются для хранения настроек, которые могут быть прочитаны и изменены текстовыми редакторами.
- Наиболее распространенные форматы:
 - INI: Содержит секции и параметры, поддерживает простое чтение и запись.
 - XML: Более сложный и структурированный формат, поддерживающий иерархическую организацию данных.
 - JSON: Легкий формат обмена данными, часто используется в веб-приложениях и современных системах.

3. Работа с реестром:

- Для работы с реестром используются утилиты, такие как regedit и командные утилиты, такие как reg.exe.
- Основные операции включают создание, изменение, удаление ключей, а также экспорт и импорт частей реестра.

4. Значение конфигурационных файлов:

- Позволяют приложениям сохранять пользовательские настройки и системные параметры, обеспечивая простоту управления.
- Часто используются для настраиваемых элементов программ, таких как параметры интерфейса и подключаемых модулей.

Задания для выполнения студентами

1. Изучение структуры реестра:

- Изучите и опишите структуру реестра Windows, включая основные корневые ключи и их назначение. Подготовьте схему для наглядности.

2. Работа с реестром:

- Откройте редактор реестра (regedit) и выполните следующие операции:
 - Найдите ключ, относящийся к программному обеспечению, установленному на вашем компьютере.
 - Создайте новый ключ и добавьте к нему значение. Опишите процесс.

3. Редактирование конфигурационного файла:

- Найдите и отредактируйте конфигурационный файл (например, config.ini или settings.xml) в любом установленном вами приложении. Измените одну из настроек и опишите процесс.
4. Создание собственного конфигурационного файла:
 - Напишите свой конфигурационный файл в формате INI и XML, включив минимальный набор параметров для воображаемого приложения. Объясните, как эти параметры будут использоваться.
 5. Изучение командной строки:
 - Используйте командную строку для выполнения операций с реестром (например, создать, изменить и удалить ключ). Опишите команды, которые использовали, и их результаты.

Контрольные вопросы по теме

1. Что такое реестр в Windows и для чего он используется?
2. Перечислите основные корневые ключи реестра и их назначение.
3. Каковы отличия между INI, XML и JSON конфигурационными файлами?
4. Какие операции можно выполнять с реестром через утилиты Windows?
5. Какова структура конфигурационного файла INI?

Самостоятельная работа № 12 (1 час)

Тема: Устройство мобильных операционных систем.

Цель работы: Углубление понимания устройства мобильных операционных систем и их особенностей.

Краткие сведения из теории

Мобильные операционные системы (МОС) — это специализированные операционные системы, разработанные для управления и оптимизации работы мобильных устройств, таких как смартфоны и планшеты. Они обеспечивают взаимодействие между аппаратным обеспечением и приложениями, а также предлагают уникальные функции, оптимизированные для сенсорных интерфейсов и мобильных условий.

1. Основные характеристики мобильных ОС:
 - Оптимизированная работа с энергоэффективностью.
 - Поддержка сенсорных экранов и жестов.
 - Возможности работы в условиях ограниченной вычислительной мощности и памяти.
 - Интеграция с сетевыми сервисами и облачными решениями.
2. Структура мобильных ОС:
 - Ядро: Отвечает за управление ресурсами устройства, управление памятью, процессами и взаимодействие с аппаратным обеспечением.
 - Системные вызовы: Позволяют приложениям взаимодействовать с ядром.
 - Интерфейсы: Поддержка различных средств разработки (SDK), инструменты для создания пользовательских интерфейсов.
3. Примеры мобильных ОС:
 - Android: Наиболее популярная мобильная ОС, основанная на ядре Linux, с открытым исходным кодом.
 - iOS: Операционная система для устройств Apple, известная своей безопасностью и закрытой экосистемой.
 - Windows Mobile: Операционная система от Microsoft (хотя сейчас уже не поддерживается).
 - KaiOS: Легковесная ОС для недорогих телефонов с поддержкой приложений.
4. Архитектура мобильных ОС:
 - Приложения: Пользовательские приложения, работающие на верхнем уровне.
 - Платформа: Слой, который обеспечивает взаимодействие между приложениями и ядром (например, Android Runtime).
 - Драйверы: Обеспечивают взаимодействие с аппаратным обеспечением, такими как сенсоры, GPS и камеры.

Задания для выполнения студентами

1. Сравнительный анализ мобильных ОС:
 - Подготовьте таблицу, в которой сравните Android и iOS по различным критериям: архитектура, возможности кастомизации, база приложений, безопасность.
2. Изучение архитектуры Android:
 - Напишите краткий отчет о архитектуре Android, описывая основные компоненты: ядро Linux, библиотеки, Android Runtime и приложение.
3. Разработка простого приложения:
 - Создайте простое приложение (например, "Hello World") на платформе Android или iOS. Опишите процесс разработки и настройки среды.
4. Исследование аспектов безопасности:

- Изучите механизмы безопасности, реализованные в Android и iOS. Напишите краткое сопоставление, уделяя внимание таким аспектам как разрешения и защищенные среды выполнения.
5. Мобильные ОС для IoT:
- Исследуйте один из мобильных операционных систем, используемых в устройствах IoT (например, KaiOS). Опишите его архитектуру и область применения.

Контрольные вопросы по теме

1. Каковы ключевые особенности мобильных операционных систем?
2. В чем отличие архитектуры Android от iOS?
3. Какие системы вызовов используются в мобильных операционных системах, и как они работают?
4. Как мобильные ОС справляются с управлением памятью и производительностью?
5. Какие механизмы защиты данных и приложений существуют в современных мобильных ОС?

Самостоятельная работа № 13 (1 час)

Тема: Модели операционных систем.

Цель работы: Углубить понимание моделей и структур операционных систем, их переосмыслении и современных тенденций в проектировании.

Краткие сведения из теории

Операционные системы (ОС) — это программное обеспечение, управляющее аппаратными ресурсами компьютера и обеспечивающее выполнение прикладных программ. Они могут быть классифицированы по различным моделям и архитектуре.

1. Модели операционных систем:

- Монолитные ОС: Все компоненты (управление процессами, файлами, памятью) находятся в одном ядре, что обеспечивает высокую производительность, но снижает надежность и удобство.
- Микроядра: Основное ядро системы минимально и выполняет лишь базовые функции, такие как управление памятью и процессами. Вся остальная функциональность реализуется в пользовательских пространствах, что повышает надежность, но может ухудшить производительность.
- Гибридные модели: Сочетают принципы монолитных и микроядерных систем, используя модули для обеспечения расширяемости и эффективности.

2. Структуры операционных систем:

- Управление процессами: Отвечает за создание, планирование, выполнение и завершение процессов.
- Управление памятью: Управляет выделением и освобождением памяти, поддерживает виртуальную память.
- Файловая система: Обеспечивает хранение и управление данными на дисках, поддерживает доступ к файлам.
- Управление устройствами: Обеспечивает взаимодействие с аппаратными устройствами через драйверы.

3. Виды ядер:

- Монолитное ядро: Весь код ОС работает в привилегированном режиме, обеспечивая высокий уровень производительности.
- Микроядро: Содержит лишь основные компоненты; предоставляет возможность запустить дополнительные функции в пользовательском пространстве.
- Гибридное ядро: Подход, сочетающий элементы и одного, и другого, предоставляя баланс между производительностью и надежностью.

4. Экзоядро:

- Это концепция, при которой ОС структурирована так, что большинство функций, не относящихся к базовым задачам, выполняются в пользовательском пространстве. Экзоядро обеспечивает большую гибкость и модульность, облегчая добавление новых функций.

Задания для выполнения студентами

1. Сравнительный анализ моделей ОС:

- Подготовьте таблицу, в которой сравните монолитные, микроядерные и гибридные операционные системы по производительности, надежности и структуре. Приведите примеры ОС для каждой модели.

2. Исследование структуры ОС:

- Выберите одну известную операционную систему (например, Linux или Windows) и напишите отчет о ее структуре, описывая, как реализованы управление процессами, памятью и файловая система.

3. Разработка простого экзоядра:
 - Создайте концептуальную схему экзоядра, описывающую разделение функций между ядром и пользовательскими модулями. Какие функции можно вынести в пользовательское пространство?
4. Сравнение ядер:
 - Напишите краткий обзор различий между монолитным, микроядром и гибридным ядром. Какой из типов ядер вы бы выбрали для разработки новой ОС и почему?
5. Практическое задание:
 - Исследуйте, как ваша ОС загружает ядро при запуске. Напишите шаги и процессы, которые происходят во время загрузки.

Контрольные вопросы по теме

1. Какие модели операционных систем существуют и каковы их основные характеристики?
2. Какова основная функция микропроцессоров, и в чем их преимущества и недостатки?
3. Каковы ключевые компоненты структуры операционной системы?
4. В чем разница между монолитным и микроядерным подходами к проектированию ОС?
5. Как экзоядро способствует модульности и гибкости операционных систем?

Самостоятельная работа № 14 (1 час)

Тема: Процессы.

Цель работы: Понять концепцию процессов и их взаимодействие в системах.

Краткие сведения из теории

Процесс — это экземпляр программы, который выполняется на компьютере, обладая собственным адресным пространством, данными, состоянием и ресурсами. Процесс является основной единицей работы операционной системы и отвечает за управление ресурсами.

1. Создание процесса:
 - Создание нового процесса обычно происходит через системный вызов `fork()`, который создает копию родительского процесса. После создания новый процесс может выполнять разные действия с помощью системных вызовов, например, `exec()` для загрузки новой программы.
2. Завершение процесса:
 - Процесс может завершиться либо успешно (с помощью `exit()`), либо с ошибкой. Завершение процесса приводит к освобождению всех выделенных ресурсом ресурсов, и ОС меняет статус процесса на "завершен".
3. Иерархии процессов:
 - Процессы могут иметь родительские и дочерние отношения, формируя иерархию. Каждый процесс имеет уникальный идентификатор (PID) и идентификатор родительского процесса (PPID), что позволяет отслеживать отношения между ними.
4. Состояния процессов:
 - Процессы могут находиться в различных состояниях:
 - Новый: процесс создается.
 - Готовый: процесс готов к выполнению, но ожидает выделения CPU.
 - Выполняющийся: процесс в данный момент выполняется.
 - Ожидающий: процесс ожидает завершения ввода-вывода или другого события.
 - Завершенный: процесс закончился.
5. Контекст процесса:
 - Контекст процесса включает в себя информацию о состоянии процесса, его регистры, стек, указатели на память и информацию о планировании. Контекст сохраняется и восстанавливается при переключении между процессами.
6. Дескриптор процесса:
 - Дескриптор процесса — это структура данных, содержащая всю необходимую информацию о процессе для операционной системы, включая PID, состояние, приоритет, адресное пространство и другие параметры.

Задания для выполнения студентами

1. Анализ создания процесса:
 - Напишите программу на C, использующую `fork()`, которая создает несколько дочерних процессов. Каждый дочерний процесс должен выводить свой PID и PPID.
2. Реализация иерархии процессов:
 - Создайте программу, показывающую иерархию процессов. Используйте `getppid()` для отображения родительского процесса и создайте несколько уровней дочерних процессов.
3. Изучение состояний процессов:

- Используйте команду `ps` или `top`, чтобы наблюдать за состоянием процессов в вашей системе. Подготовьте отчет о различных состояниях активных процессов.
- 4. Сохранение и переключение контекста:
 - Разработайте простую модель, где вы эмулируете переключение контекста между двумя процессами с помощью временных задержек (например, через `sleep()`). Определите, как меняется состояние каждого процесса.
- 5. Исследование межпроцессорного взаимодействия (IPC):
 - Напишите пример, демонстрирующий одно из средств IPC (например, каналы или семафоры) для обмена данными между двумя процессами.

Контрольные вопросы по теме

1. Что такое процесс и какие его основные характеристики?
2. Как происходит создание и завершение процесса в операционной системе?
3. Назовите и опишите различные состояния процессов.
4. Какова роль контекста процесса и что он включает в себя?
5. Чем отличается дескриптор процесса от контекста процесса?
6. Как организована иерархия процессов и какие примеры можно привести?

Самостоятельная работа № 15 (1 час)

Тема: Потоки.

Цель работы: Понять концепции потоков, их реализацию и управление в операционных системах.

Краткие сведения из теории

Потоки — это легковесные процессы, которые позволяют выполнять несколько операций параллельно в пределах одного процесса. Потоки делят свою память и ресурсы, что делает их менее затратными по сравнению с полноправными процессами.

1. Определение потоков:
 - Поток — это минимальная единица обработки, которая может выполняться отдельно. Каждый поток имеет свое собственное состояние, стек и контекст, но разделяет память и ресурсы с другими потоками того же процесса.
2. Классическая модель потоков:
 - В классической модели потоки могут быть:
 - Многие к одному: несколько пользовательских потоков сопоставляются одному потоку ядра. Преимущество — меньше накладных расходов, недостаток — блокировка одного потока может блокировать весь процесс.
 - Один к одному: каждый пользовательский поток соответствует своему потоку ядра. Преимущество — высокая параллельность, недостаток — большой объем системных ресурсов.
 - Многие ко многим: множество пользовательских потоков могут быть сопоставлены с множеством потоков ядра. Это наиболее гибридная модель, позволяющая эффективно использовать все доступные ресурсы.
3. Реализация потоков:
 - В пользовательском пространстве: потоки управляются библиотеками на уровне приложения, что позволяет избежать накладных расходов при взаимодействии с системой. Примеры — Pthreads и user-level threads.
 - В ядре: потоки осуществляются средствами операционной системы, каждый поток управляется ядром, что позволяет обеспечивать эффективное распределение CPU и управление приоритетами.
 - Гибридная реализация: использует как пользовательскую, так и ядерную модели, обеспечивая баланс между производительностью и управляемостью.
4. Всплывающие потоки (называемые также "потоки в ожидании"):
 - Потоки, которые временно блокируются в ожидании ресурсов (например, ввода-вывода) и могут "всплывать" или быть активированными, когда ресурсы становятся доступны. Это позволяет эффективно использовать процессорное время.

Задания для выполнения студентами

1. Исследование моделей потоков:
 - Подготовьте сравнительный анализ трех моделей потоков (многие к одному, один к одному, многие ко многим). Укажите плюсы и минусы каждой модели.
2. Программирование потоков:
 - Напишите программу на C с использованием Pthreads, которая создает несколько потоков, каждый из которых выполняет вычисления (например, вычисление суммы чисел от 1 до N). Сравните время выполнения программы с использованием одного потока.
3. Выявление блокировок:

- Напишите пример, демонстрирующий проблемы блокировки в модели "многие к одному". Показать, как один поток может заблокировать выполнение остальных.
4. Симуляция всплывающих потоков:
 - Реализуйте простую модель с использованием Python с потоками, где один поток выполняет длительную операцию, а другой — вводится в блокировку. Показать, как система обрабатывает эти ситуации (используйте библиотеку threading).
 5. Доклад о реализации:
 - Подготовьте краткий доклад о методах реализации пользовательских и ядерных потоков в вашей операционной системе, включая примеры и их применения.

Контрольные вопросы по теме

1. Что такое поток и как он отличается от процесса?
2. Опишите три классические модели потоков. Каковы их преимущества и недостатки?
3. Какие аспекты управления потоками принадлежат операционной системе?
4. Объясните, что такое "всплывающие потоки" и какую роль они играют в управлении ресурсами.
5. Каковы основные параметры производительности, которые нужно учитывать при работе с потоками?

Самостоятельная работа № 16 (1 час)

Тема: Управление процессами в Linux.

Цель работы: Углубить понимание управления процессами в Linux и развить навыки программирования и администрирования.

Краткие сведения из теории

Управление процессами в ОС Linux включает в себя создание, выполнение, завершение и взаимодействие между процессами. Процессом считается любое выполняющееся приложение, которое получает системные ресурсы для выполнения своих задач.

1. Основные понятия:

- **Процесс:** Исполняемая программа, имеющая свое собственное адресное пространство.
- **PID (Идентификатор процесса):** Уникальный номер, присвоенный каждому процессу.
- **PPID (Идентификатор родительского процесса):** Номер процесса, который создал текущий процесс.
- **Состояния процессов:** Новая, готовая, выполняющаяся, ожидающая и завершенная.

2. Системные вызовы для управления процессами:

- **fork():** Создает новый процесс, копируя существующий.
- **exec():** Заменяет текущий процесс новым образом.
- **wait():** Ожидает завершения дочернего процесса.
- **kill():** Отправляет сигнал процессу, например, для его завершения.

3. Управление процессами:

- Процессы в Linux могут быть фоновые и передние. Пользователь может управлять ими через команды в терминале.
- Приоритеты процессов: можно изменять с помощью команд `nice` и `renice`.
- Использование `ps`, `top` и `htop` для мониторинга и управления активными процессами.

Задания для выполнения студентами

1. Создание и завершение процесса:

- Напишите простую программу на C, которая создает новый процесс с помощью `fork()`, выполняет команду `ls`, а затем завершает процесс с помощью `exit()`.

2. Мониторинг процессов:

- Используя команду `ps`, составьте отчет о текущих выполняющихся процессах в вашей системе. Укажите PID, PPID, время выполнения и состояние.

3. Использование системных вызовов:

- Напишите программу на C, которая использует `fork()` и `exec()` для запуска программы `date`. Процесс должен отобразить текущее время и завершиться.

4. Управление процессами с помощью команд:

- Используйте команды `top` и `htop` для мониторинга ресурсов, используемых разными процессами. Сделайте скриншот и прокомментируйте, какие процессы занимают наибольшее количество ресурсов.

5. Сигналы и управление процессами:

- Напишите скрипт на Bash, который запускает фоновый процесс, затем посылает ему сигнал остановки (`SIGSTOP`) и последующее восстановление (`SIGCONT`).

Контрольные вопросы по теме

1. Какие системные вызовы вы можете использовать для создания нового процесса в Linux?
2. В чем разница между `fork()` и `exec()`?
3. Как работают сигналы в Linux и как их можно использовать для управления процессами?
4. Как можно изменить приоритет процесса и зачем это может быть нужно?
5. Опишите, какие команды можно использовать для мониторинга состояния процессов в Linux. Какие данные они предоставляют?

Самостоятельная работа № 17 (1 час)

Тема: Пользовательские скрипты в Unix.

Цель работы: Освоить принципы создания и использования пользовательских скриптов.

Краткие сведения из теории

Пользовательские скрипты в Unix — это программы, написанные на скриптовых языках, таких как Bash, Perl, Python и другие, которые автоматизируют выполнение задач в командной оболочке. Скрипты позволяют пользователям выполнять сложные операции, комбинируя несколько команд, и упрощают взаимодействие с системой.

1. Общие принципы создания скриптов:
 - Скрипт начинается с шеебанга (`#!/bin/bash`), который указывает интерпретатор.
 - Команды записываются по строчно, и каждая команда выполняется последовательно.
 - Скрипты могут принимать аргументы, обрабатывая ввод пользователя.
 - Использование переменных, циклов, условных операторов и функций делает скрипты более мощными и гибкими.
2. Применение пользовательских скриптов:
 - Автоматизация рутинных задач (резервное копирование, очистка, обновление).
 - Управление процессами и настройка системы.
 - Разработка пользовательских утилит для специфических задач.
3. Основные конструкции:
 - Переменные: Объявляются и присваиваются значения с помощью `VAR_NAME=value`.
 - Условные операторы: `if`, `then`, `else`.
 - Циклы: `for`, `while`.
 - Функции: Определяются с помощью `function_name() {}`.

Задания для выполнения студентами

1. Создание простого скрипта:
 - Напишите скрипт, который выводит на экран "Hello, World!" и сохраняет результат в файл `output.txt`.
2. Скрипт с аргументами:
 - Создайте скрипт, который принимает два аргумента — имя файла и текст. Скрипт должен записывать текст в указанный файл, создавая его, если он не существует.
3. Цикл и условие:
 - Напишите скрипт, который принимает на вход число и выводит все четные числа от 1 до этого числа. Используйте условный оператор и цикл.
4. Автоматизация задач:
 - Создайте скрипт для резервного копирования указанной директории. Скрипт должен использовать команду `tar` для создания архива и сохранять его в папке `~/backups`.
5. Документация скриптов:
 - Напишите комментарии к каждому из созданных вами скриптов, поясняющие, что делает каждая часть вашего кода.

Контрольные вопросы по теме

1. Что такое пользовательские скрипты и каковы их основные преимущества?
2. Какие конструкции языка Bash вы использовали при написании своих скриптов?
3. Как работает оператор `if` в Bash? Приведите пример.
4. Как можно обрабатывать аргументы командной строки в скриптах Unix?
5. Почему важно документировать скрипты и какие комментарии лучше использовать?

Самостоятельная работа № 18 (2 часа)

Тема: Управление памятью.

Цель работы: Понять управление памятью в операционных системах, влияние на производительность системы.

Краткие сведения из теории

Управление памятью — это ключевая задача операционных систем, которая заключается в эффективном распределении и использовании ресурсов оперативной памяти для выполнения процессов.

1. Основное управление памятью:

- Основное управление памятью включает в себя выделение, освобождение и защиту памяти, а также обеспечение изоляции процессов. Основные схемы управления памятью:
 - Память фиксированной длины: Простое, но неэффективное распределение памяти, при котором каждому процессу выделяется фиксированный объем.
 - Динамическое распределение памяти: Память выделяется по мере необходимости, что позволяет оптимизировать использование памяти.

2. Подкачка (Paging):

- Подкачка — это технология, позволяющая процессам использовать память, выходящую за пределы физической памяти. Операционная система разбивает память на небольшие блоки (страницы) и перемещает их между оперативной памятью и диском. Это позволяет более эффективно использовать память, но может приводить к снижению производительности из-за частого обращения к диску.

3. Виртуальная память:

- Виртуальная память — это метод, который позволяет использовать больше памяти, чем физически доступно, создавая иллюзию большой адресной пространства. Каждый процесс получает свой собственный адресный space, который может быть значительно больше, чем физическая память. Это достигается путем использования подкачки и других методов управления памятью, таких как свопинг.

Задания для выполнения студентами

1. Сравнительный анализ:

- Подготовьте таблицу, в которой сравните основные схемы управления памятью (фиксированное и динамическое распределение). Укажите плюсы и минусы каждой схемы.

2. Технология подкачки:

- Опишите процесс подкачки на примере. Как происходит обращение к данным в памяти, когда используется подкачка? Объясните, как минимизировать эффект свопинга.

3. Виртуальная память:

- Напишите отчет о принципах работы виртуальной памяти, объясните, как она реализуется в современных операционных системах. Укажите, как виртуальная память влияет на производительность системы.

4. Решение задачи:

- Создайте небольшую программу, которая демонстрирует использование памяти. Проанализируйте, сколько памяти выделяется для разных объектов и как это выглядит в системе.

5. Кейс-стадия:

- Исследуйте, как ваша операционная система (Windows, Linux и т.д.) реализует виртуальную память. Подготовьте краткое резюме о найденной информации.

Контрольные вопросы по теме

1. Что такое управление памятью и почему оно важно в операционных системах?
2. Каковы основные функции управления памятью в операционной системе?
3. Объясните, что такое подкачка, и как она работает.
4. Что такое виртуальная память и какие преимущества она предоставляет?
5. Какие основные проблемы могут возникать из-за использования подкачки и виртуальной памяти?

Самостоятельная работа № 19 (2 часа)

Тема: Системные вызовы.

Цель работы: Углубить знания о системных вызовах и реализации ввода-вывода в операционных системах.

Краткие сведения из теории

Системные вызовы — это интерфейс между приложениями и операционной системой, который позволяет программам запрашивать услуги у ОС. В контексте ввода-вывода (I/O) системные вызовы играют ключевую роль в взаимодействии программ с аппаратными устройствами.

1. Системные вызовы ввода-вывода:
 - Системные вызовы для ввода-вывода позволяют программе осуществлять операции над файлами и устройствами. Основные вызовы включают:
 - `read()`: Чтение данных из файла или устройства.
 - `write()`: Запись данных в файл или устройство.
 - `open()`: Открытие файла или устройства для работы.
 - `close()`: Закрытие файла после завершения работы.
 - `ioctl()`: Управление устройствами на более низком уровне.
2. Реализация ввода-вывода в операционных системах:
 - Операционные системы используют различные техники для управления вводом-выводом:
 - Прерывание: Устройство инициирует прерывание, оповещая ЦП о завершении операции ввода-вывода.
 - Опрашивание: Процесс периодически проверяет статус устройства на предмет готовности.
 - Директ-метод ввод-вывод: Устройства имеют прямой доступ к памяти без участия процессора, что повышает производительность.
 - Буферизация: Данные временно хранятся в буферах, что помогает избежать блокировок и увеличивает скорость передачи данных.

Задания для выполнения студентами

1. Исследование системных вызовов:
 - Подготовьте краткий отчет о каждом из системных вызовов ввода-вывода (`read`, `write`, `open`, `close`). Укажите, какую информацию они принимают и возвращают, а также их применение в реальных сценариях.
2. Графическое представление реализации I/O:
 - Создайте блок-схему, иллюстрирующую процесс выполнения операции ввода-вывода, начиная с системного вызова и заканчивая завершением операции.
3. Сравнение методов ввода-вывода:
 - Проведите анализ различных методов реализации ввода-вывода (прерывания, опрашивание, директ-метод и буферизация). Сравните их по эффективности, сложности и применимости.
4. Решение задачи программирования:
 - Напишите простую программу на любом языке, реализующую операции чтения из файла и записи в файл с использованием системных вызовов. Объясните, как ваша программа работает и какие системные вызовы используются.
5. Изучение устройства ввода-вывода:
 - Исследуйте один тип устройства ввода-вывода (например, принтер, клавиатура, диск) и опишите, как операционная система взаимодействует с этим устройством через системные вызовы.

Контрольные вопросы по теме

1. Что такое системные вызовы и какова их роль в операционных системах?
2. Приведите примеры системных вызовов, связанных с вводом-выводом.
3. Каковы основные методы реализации ввода-вывода в операционных системах?
4. В чем разница между опрашиванием и прерываниями в контексте ввода-вывода?
5. Почему важна буферизация при работе с устройствами ввода-вывода?

Самостоятельная работа № 20 (2 часа)

Тема: Выгружаемые и невыгружаемые ресурсы.

Цель работы: Понимание управления ресурсами в операционных системах и проблем взаимоблокировок.

Краткие сведения из теории

Ресурсы в контексте операционных систем представляют собой физические или логические элементы, которые используются процессами для выполнения своих задач. Ресурсы могут быть различными: память, процессорное время, устройства ввода-вывода и т.д.

1. Выгружаемые и невыгружаемые ресурсы:

- **Выгружаемые ресурсы:** Это ресурсы, которые могут быть освобождены после использования, например, оперативная память, процессорное время и устройства ввода-вывода. Процесс может временно возратить их, чтобы другие процессы могли их использовать.
- **Невыгружаемые ресурсы:** Это ресурсы, которые не могут быть освобождены, пока процесс не завершится. Например, уникальные устройства (принтеры, сканеры) или замки в многопоточном программировании. Пока один процесс не завершится, другие не смогут получить доступ к таким ресурсам.

2. Условия возникновения ресурсных взаимоблокировок: Взаимоблокировка происходит, когда несколько процессов ожидают освобождения ресурсов, которые уже не доступны, что приводит к ситуации, когда ни один из процессов не может продолжить работу. Чтобы возникла взаимоблокировка, должны выполняться следующие четыре условия:

- **Взаимное исключение:** По крайней мере, один ресурс должен быть держим в режиме недоступном (нельзя одновременно предоставить нескольким процессам).
- **Удержание и ожидание:** Процесс должен удерживать как минимум один ресурс и ожидать получения других ресурсов, которые в данный момент занимают другие процессы.
- **Нет высвобождения:** Удерживаемые ресурсы не могут быть принудительно забраны у процесса, они могут быть освобождены только по завершению процесса.
- **Циклическое ожидание:** Должен существовать цикл процессов, в котором каждый процесс ожидает ресурс, который удерживается следующим процессом в цикле.

Задания для выполнения студентами

1. Классификация ресурсов:

- Составьте таблицу, в которой перечислите различные типы ресурсов, выделите между ними выгружаемые и невыгружаемые ресурсы. Приведите примеры.

2. Сценарий взаимоблокировки:

- Придумайте сценарий, в котором происходит взаимоблокировка между тремя процессами. Опишите его шаги и укажите, какие ресурсы задействованы.

3. Анализ методов разрешения взаимоблокировок:

- Изучите методы предотвращения и разрешения взаимоблокировок (например, алгоритмы Фуллера, банкира). Подготовьте краткий отчет о каждом из методов, указав их преимущества и недостатки.

4. Практическое задание на программирование:

- Напишите небольшую программу (на любом языке), которая иллюстрирует взаимоблокировку. Объясните, как она работает, и как можно избежать этой проблемы.

5. Исследование функций операционной системы:

- Проанализируйте, как современные операционные системы (например, Windows, Linux) управляют ресурсами. Обратите особое внимание на то, как они избавляются от взаимоблокировок.

Контрольные вопросы по теме

1. Что такое ресурсы, используемые в операционных системах?
2. Приведите примеры выгружаемых и невыгружаемых ресурсов.
3. Какие условия необходимы для возникновения взаимоблокировок?
4. Как можно предотвратить взаимоблокировку в системах?
5. В чем разница между взаимоблокировкой и просто высокой загрузкой ресурсов?

Самостоятельная работа № 21 (2 часа)

Тема: Сегментация памяти.

Цель работы: Изучить информацию о сегментации памяти и понять её значение в контексте операционных систем.

Краткие сведения из теории

Сегментация памяти — это техника управления памятью, при которой память разделяется на логические сегменты, каждый из которых представляет собой отдельную часть программы или данные. В отличие от страничной памяти, где память разбивается на фиксированные странички, сегментация использует переменные размеры сегментов, что более эффективно отражает структуру программы.

1. Основные понятия:

- **Сегмент:** Логическое разделение памяти, которое может содержать код, данные или стек программы. Каждый сегмент имеет уникальный номер и длину.
- **Таблица сегментов:** Структура данных, которая хранит информацию о каждом сегменте, включая его базовый адрес и длину.
- **Адресация:** В сегментации адресация происходит с использованием двух составляющих: номера сегмента и смещения внутри сегмента.

2. Преимущества сегментации:

- **Логическая организация памяти:** Сегменты могут структурировано соответствовать логическим элементам программы (например, функции, глобальные переменные).
- **Упрощение управления памятью:** Позволяет более гибкое выделение и освобождение памяти.
- **Изоляция процессов:** Обеспечивает защиту данных между процессами за счёт разделения сегментов.

3. Недостатки сегментации:

- **Фрагментация:** Сегментация может привести к внешней фрагментации, в результате чего в памяти остаются свободные области, которые не могут быть использованы.
- **Сложность управления:** Требуются дополнительные данные для отслеживания сегментов, что нагружает систему.

Задания для выполнения студентами

1. Исследование механизмов сегментации:

- Найдите и изучите различные системы, использующие сегментацию памяти (например, Intel 80386). Сделайте краткий обзор их работы и представьте примеры сегментов.

2. Создание таблицы сегментов:

- Выберите простую программу и создайте таблицу сегментов, описывающую её структуру. Укажите размеры и адреса для каждого сегмента.

3. Анализ преимуществ и недостатков:

- Напишите краткий анализ преимуществ и недостатков использования сегментации по сравнению с другими методами управления памятью, такими как страничная адресация.

4. Моделирование фрагментации:

- Смоделируйте процесс выполнения программы с использованием сегментации. Опишите, как может возникнуть внешняя фрагментация при выделении и освобождении сегментов.

5. Сравнение с страничной памятью:

- Подготовьте сравнительную таблицу, в которой выделите ключевые различия между сегментацией и страничной памятью, особенно в контексте управления памятью и производительности.

Контрольные вопросы по теме

1. Что такое сегментация памяти и какую роль она играет в операционных системах?
2. Какова структура таблицы сегментов и какую информацию она содержит?
3. В чем заключается основное отличие между сегментацией и страничной адресацией?
4. Какие проблемы возникают при использовании сегментации, и как их можно минимизировать?
5. Как сегментация способствует изоляции процессов в операционной системе?

Самостоятельная работа № 22 (2 часа)

Тема: Основные понятия безопасности, классификация угроз.

Цель работы: Понять основные аспекты безопасности информации и классификацию угроз, что является важным шагом к безопасному функционированию информационных систем.

Краткие сведения из теории

Безопасность информационных систем — это комплекс мер, механизмов и технологий, направленных на защиту информации и ресурсов от несанкционированного доступа, утечек, повреждений и уничтожения. Важными аспектами безопасности являются конфиденциальность, целостность и доступность данных.

1. Основные понятия безопасности:

- **Конфиденциальность:** Защита информации от несанкционированного доступа. Обеспечивается средствами аутентификации и шифрования.
- **Целостность:** Гарантия того, что информация не была изменена или уничтожена без разрешения. Используются методы контроля целостности, такие как хэширование.
- **Доступность:** Обеспечение возможности доступа к данным и системам для авторизованных пользователей. Включает стратегии резервного копирования и защиты от отказов.

2. Классификация угроз:

- **Угрозы программного обеспечения:**
 - Вредоносные программы: Вирусы, черви, трояны.
 - Уязвимости ПО: Ошибки, которые могут быть использованы злоумышленниками для атаки на систему.
- **Угрозы на уровне сети:**
 - Атаки через сеть: DDoS-атаки, подслушивание трафика.
 - Социальная инженерия: Методы манипуляции людьми для получения конфиденциальной информации.
- **Физические угрозы:**
 - Потеря оборудования: Кража или уничтожение устройств, содержащих данные.
 - Природные катастрофы: Землетрясения, потопа, пожары, способные повредить физические ресурсы.

Задания для выполнения студентами

1. **Исследование угроз программного обеспечения:**
 - Выберите один вид вредоносного ПО (вирус, червь или троян). Исследуйте его механизм действия, последствия заражения и методы защиты от него. Подготовьте краткий отчет.
2. **Анализ уязвимостей:**
 - Составьте список наиболее распространённых уязвимостей ПО (например, SQL-инъекции, XSS). Укажите последствия их использования и предложите способы устранения.
3. **Практическое задание по социальной инженерии:**
 - Изучите методы социальной инженерии. Проведите опрос среди знакомых, чтобы выяснить, насколько они осведомлены о таких угрозах и могут ли распознать попытку манипуляции.
4. **Изучение методов защиты конфиденциальности:**
 - Исследуйте методы шифрования данных. Сравните симметричное и асимметричное шифрование с точки зрения удобства и безопасности. Подготовьте сравнительную таблицу.

5. Сценарий реагирования на угрозу:

- Разработайте сценарий реагирования на гипотетическую угрозу (например, утечка данных). Укажите шаги, которые необходимо предпринять для минимизации ущерба и защиты системы.

Контрольные вопросы по теме

1. Что включает в себя понятие безопасности информации?
2. Какие основные методы используются для обеспечения конфиденциальности данных?
3. В чем разница между физическими и сетевыми угрозами?
4. Каковы основные признаки заражения компьютера вредоносными программами?
5. Какие существуют способы защиты против атак социальной инженерии?

Самостоятельная работа № 23 (2 часа)

Тема: Технологии безопасности, механизмы защиты.

Цель работы: Понять технологии безопасности, механизмы защиты и основы построения надежных систем.

Краткие сведения из теории

Безопасность информационных систем — это совокупность мер, направленных на защиту информации от несанкционированного доступа, изменения, повреждения или уничтожения. Этот аспект включает в себя как технологии, так и процессы, обеспечивающие конфиденциальность, целостность и доступность данных.

1. Базовые технологии безопасности:

- **Шифрование:** Используется для защиты данных путем их преобразования в нечитабельный вид. Подразделяется на симметричное (один ключ для шифрования и расшифровки) и асимметричное (пара ключей: публичный и приватный).
- **Аутентификация:** Процесс проверки идентичности пользователя или системы. Может осуществляться через пароли, биометрическую идентификацию, токены и сертификаты.
- **Авторизация:** Определяет права доступа пользователя к ресурсам системы после его аутентификации.

2. Механизмы защиты:

- **Файрволы:** Аппаратные или программные средства, ограничивающие или разрешающие трафик между сетями на основе предустановленных правил.
- **Антивирусные программы:** Обнаруживают и устраняют вредоносные программы, защищая системы от нежелательных атак.
- **Обнаружение и предотвращение вторжений (IDS/IPS):** Системы мониторинга, которые отслеживают подозрительную активность и реагируют на неё.
- **Резервное копирование:** Защита данных путём регулярного создания их копий на внешних носителях.

3. Надежные системы:

- **Надежная система** — это система, способная обеспечивать безопасность и гарантировать защиту данных от угроз. Ключевые элементы:
 - **Принципы минимизации привилегий:** пользователи и программы должны иметь только те права, которые необходимы для выполнения их функций.
 - **Модульность:** системы должны быть построены из независимых модулей, которые можно обновлять и проверять по отдельности.
 - **Регулярное обновление и патчинг:** для закрытия уязвимостей в программном обеспечении.

Задания для выполнения студентами

1. Исследование шифрования:

- Изучите методы симметричного и асимметричного шифрования. Напишите краткий отчет о каждом методе, его преимуществах и недостатках.

2. Анализ фаерволов:

- Настройте программный фаервол на вашем компьютере и проведите эксперимент, блокируя определенные порты. Опишите процесс настройки и результаты.

3. Тестирование антивирусного ПО:

- Установите антивирусное программное обеспечение и проведите полное сканирование системы. Запишите обнаруженные угрозы и действия, предпринятые антивирусом.
- 4. Применение принципов минимизации привилегий:
 - Рассмотрите свою учетную запись на компьютере и определите, какие улучшения в безопасности можно внести, изменив права доступа, и как это влияет на работу.
- 5. Разработка плана безопасности:
 - Создайте упрощенный план безопасности для малой организации, включающий политику резервного копирования, защиты данных и реагирования на инциденты.

Контрольные вопросы по теме

1. Каковы основные цели безопасности информационных систем?
2. В чем заключается отличие между аутентификацией и авторизацией?
3. Какие типы фаерволов существуют, и как они работают?
4. Что такое IDS и IPS, и в чем их основные различия?
5. Как принципы минимизации привилегий помогают повысить безопасность системы?

Самостоятельная работа № 24 (2 часа)

Тема: Резервное копирование и восстановление данных в Windows.

Цель работы: Понять процессы резервного копирования и восстановления данных в Windows, что является важным навыком для эффективного управления данными

Краткие сведения из теории

Резервное копирование и восстановление данных в Windows — это ключевые процессы для сохранения информации и обеспечения её целостности. Эти процессы помогают предотвратить потерю данных из-за случайного удаления, сбоев оборудования, вирусных атак и других нежелательных ситуаций.

1. Типы резервного копирования:
 - Полное резервное копирование: создаёт полную копию всех выбранных файлов и папок. Это обеспечивает полный доступ к данным при восстановлении.
 - Инкрементное резервное копирование: сохраняет изменения с последнего резервного копирования (полного или инкрементного), что позволяет экономить время и место.
 - Дифференциальное резервное копирование: сохраняет изменения, произошедшие с последнего полного резервного копирования. Это более быстрый способ восстановления, чем инкрементное.
2. Инструменты резервного копирования в Windows:
 - Резервное копирование и восстановление (Windows 7): встроенный инструмент для создания резервных копий файлов и изображений системы.
 - История файлов (Windows 8 и выше): позволяет сохранять и восстанавливать отдельные версии файлов по времени.
 - Windows Backup: более продвинутая система резервного копирования, доступная в более поздних версиях Windows.
3. Процесс восстановления:
 - Восстановление данных может выполняться из резервной копии, созданной с помощью встроенных инструментов. Важно знать, как воспользоваться этими инструментами для восстановления системы или файлов.
4. Оптимизация и планирование резервного копирования:
 - Разработка плана резервного копирования, который включает частоту выполнения, хранилище для резервных копий (внешние носители, облачные хранилища) и тип резервирования.

Задания для выполнения студентами

1. Создание резервной копии:
 - Используйте утилиту "Резервное копирование и восстановление" для создания полной резервной копии важной папки (например, "Документы"). Опишите шаги, которые вы предприняли.
2. Использование Истории файлов:
 - Активируйте и настройте Историю файлов на вашем компьютере. Сохраните несколько версий файлов и восстановите один из них. Опишите процесс.
3. Инкрементное резервное копирование:
 - Создайте инкрементное резервное копирование вашей системы с помощью доступных средств Windows. Запишите команды, если использовали PowerShell или утилиты командной строки.
4. Восстановление системы:

- Выполните восстановление системы с использованием одной из созданных резервных копий. Опишите процесс и обратите внимание на все важные моменты.
5. Разработка плана резервного копирования:
- Составьте план резервного копирования для личного компьютера, учитывающий частоту, типы резервных копий и методы хранения. Обоснуйте выбор.

Контрольные вопросы по теме

1. Какие основные шаги включает в себя процесс резервного копирования в Windows?
2. Чем отличается инкрементное резервное копирование от дифференциального?
3. Какие инструменты Windows вы можете использовать для резервного копирования и восстановления данных?
4. Какова роль облачных хранилищ в резервном копировании?
5. Какие меры предосторожности необходимо соблюдать при восстановлении данных из резервной копии?

Самостоятельная работа № 25 (2 часа)

Тема: Резервное копирование и восстановление данных в Unix.

Цель работы: Понять процессы резервного копирования и восстановления данных в Unix, что является критически важным навыком в администрировании систем.

Краткие сведения из теории

Резервное копирование (бэкап) и восстановление данных — важные процессы управления данными в операционных системах, включая Unix. Они обеспечивают сохранность информации и позволяют восстановить данные в случае утраты, повреждения или других нештатных ситуаций. Основные аспекты резервного копирования и восстановления в Unix включают:

1. Типы резервного копирования:
 - Полное: Создание полной копии всех файлов и директорий. Позволяет обеспечить полное восстановление.
 - Инкрементное: Копирует только измененные или новые файлы с последнего полного или инкрементного бэкапа. Экономит пространство и время.
 - Дифференциальное: Копирует все изменения с последнего полного резервного копирования. Позволяет быстрее восстановить данные, чем инкрементное.
2. Инструменты резервного копирования:
 - tar: Команда для создания архивов, которая может упаковывать файлы и каталоги в один файл.
 - cpio: Альтернативный инструмент для архивирования данных, позволяет создавать архивы из файловой системы.
 - rsync: Утилита для синхронизации и резервного копирования с минимальными изменениями, что экономит время.
3. Резервные копии в облаке и на внешних носителях:
 - Резервирование данных на внешних устройствах (USB, HDD) или облачных сервисах. Помогает защитить данные от физического повреждения оборудования.
4. Восстановление данных:
 - Восстановление данных может быть выполнено с помощью тех же утилит, что использовались для резервного копирования. Важно иметь план восстановления, который определяет порядок действий в случае потери данных.

Задания для выполнения студентами

1. Создание резервной копии:
 - Используйте команду tar для создания резервной копии важной директории. Опишите порядок действий и параметры, которые вы использовали.
2. Восстановление данных:
 - Восстановите данные из ранее созданного архива. Опишите процесс восстановления и любые сложности, с которыми вы столкнулись.
3. Использование rsync:
 - Практикуйтесь в использовании утилиты rsync для создания инкрементной резервной копии директории. Документируйте команды и параметры, которые вы использовали.
4. Сравнительный анализ инструментов:
 - Проведите сравнительный анализ инструментов tar, cpio и rsync. Укажите преимущества и недостатки каждого из них при резервном копировании и восстановлении данных.
5. План резервного копирования:

- Разработайте план резервного копирования для вашей персональной или учебной системы. Включите частоту резервных копий, типы резервного копирования и место хранения.

Контрольные вопросы по теме

1. Каковы основные типы резервного копирования и чем они отличаются друг от друга?
2. Какие команды и утилиты используются для резервного копирования и восстановления в Unix?
3. Почему важно проводить регулярное резервное копирование данных?
4. Каковы преимущества использования rsync перед tar для резервного копирования?
5. Каковы основные шаги в процессе восстановления данных из резервной копии?

Самостоятельная работа № 26 (2 часа)

Тема: Сетевая модель OSI.

Цель работы: Понять сетевую модель OSI и её значение в современных компьютерных сетях.

Краткие сведения из теории

Сетевая модель OSI (Open Systems Interconnection) — это концептуальная модель, описывающая, как данные передаются по компьютерным сетям. Модель OSI делится на семь уровней, каждый из которых выполняет определенные функции и служит для упрощения процесс разработки и управления сетевыми приложениями.

1. Физический уровень (Physical Layer):
 - Обеспечивает передачу необработанных битов через физические среды (кабели, радиоволны). Специфицирует электрические, механические и процедурные аспекты передач.
2. Канальный уровень (Data Link Layer):
 - Отвечает за надежную передачу данных между узлами сети. Устраняет ошибки и осуществляет доступ к среде передачи (например, Ethernet).
3. Сетевой уровень (Network Layer):
 - Обеспечивает маршрутизацию данных между сетями. Основной протокол — IP (Internet Protocol).
4. Транспортный уровень (Transport Layer):
 - Гарантирует надежную передачу данных между хостами. Предоставляет механизмы управления потоком и исправления ошибок (например, TCP и UDP).
5. Сессионный уровень (Session Layer):
 - Управляет сеансами связи между приложениями. Устанавливает, поддерживает и завершает соединения.
6. Уровень представления (Presentation Layer):
 - Отвечает за преобразование данных из формата приложения в формат, подходящий для передачи, и наоборот. Может включать шифрование и сжатие данных.
7. Прикладной уровень (Application Layer):
 - Предоставляет интерфейсы для приложений и конечных пользователей. Примеры протоколов: HTTP, FTP, SMTP.

Модель OSI помогает разработчикам и сетевым инженерам стандартизировать сетевое взаимодействие и упростить проблемы, которые могут возникать на разных уровнях.

Задания для выполнения студентами

1. Сравнительный анализ уровней модели OSI и TCP/IP:
 - Подготовьте таблицу, отображающую различия между уровнями модели OSI и моделью TCP/IP, включая названия и функции каждого уровня.
2. Изучение протоколов:
 - Выберите три протокола прикладного уровня (например, HTTP, FTP, SMTP) и исследуйте их функции, применение и взаимодействие с другими уровнями модели OSI. Подготовьте краткий отчет.
3. Практическое задание по маршрутизации:
 - Рассмотрите процесс маршрутизации IP-пакетов на сетевом уровне. Проведите анализ пути передачи данных между двумя устройствами в вашей сети (например, с помощью команды `tracert` или `tracert /r`), объясните полученные результаты.
4. Моделирование сетевой ситуации:

- Создайте простую схему сети, включающую устройства, протоколы и уровни модели OSI. Опишите, как данные проходят через каждый уровень от источника к получателю.
5. Тестирование сети:
- Используйте утилиты для диагностики сети (ping, ipconfig, ifconfig) и проведите тестирование, описав свои действия на каждом уровне модели OSI.

Контрольные вопросы по теме

1. Какие основные функции выполняет каждый уровень модели OSI?
2. Какова роль сетевого уровня в модели OSI и какие протоколы ему соответствуют?
3. В чем заключаются основные различия между транспортным уровнем TCP и UDP?
4. Как модель OSI помогает в стандартизации и решении проблем в сетях?
5. Почему важно учитывать все уровни модели OSI при проектировании сетевых приложений?

Самостоятельная работа № 27 (2 часа)

Тема: Основные протоколы передачи данных.

Цель работы: Углубить знания о протоколах передачи данных и их ключевом значении в сетевых коммуникациях.

Краткие сведения из теории

Протоколы передачи данных — это набор правил и соглашений, определяющих, как данные передаются между устройствами в компьютерных сетях. Они обеспечивают правильное форматирование, передачу, обработку и получение данных, а также отвечают за целостность и безопасность данных. Основные протоколы можно разделить на несколько категорий:

1. Протоколы канального уровня:
 - Ethernet: Широко используемый протокол для локальных сетей, работающий на основе технологии пакетной передачи.
 - Wi-Fi (802.11): Протоколы для беспроводной передачи данных, обеспечивающие связь через радиоволны.
2. Протоколы сетевого уровня:
 - IP (Internet Protocol): Основной протокол для адресации и маршрутизации пакетов данных в сети. Обеспечивает уникальную адресацию для каждого устройства.
 - ICMP (Internet Control Message Protocol): Используется для отправки ошибок и состояния сообщений, например, в утилите ping.
3. Протоколы транспортного уровня:
 - TCP (Transmission Control Protocol): Надежный протокол, обеспечивающий гарантированную доставку данных с восстановлением при ошибках и контролем потока.
 - UDP (User Datagram Protocol): Ненадежный, но более быстрый протокол, используемый для передачи данных без установления соединения (например, для потокового видео и VoIP).
4. Протоколы прикладного уровня:
 - HTTP (HyperText Transfer Protocol): Протокол для передачи веб-страниц.
 - FTP (File Transfer Protocol): Протокол для передачи файлов между клиентом и сервером.
 - SMTP (Simple Mail Transfer Protocol): Протокол для отправки электронной почты.

Эти протоколы обеспечивают взаимодействие и эффективную передачу данных в различных сетевых средах.

Задания для выполнения студентами

1. Сравнительный анализ протоколов:
 - Подготовьте сравнительную таблицу для TCP и UDP, включая их особенности, преимущества и недостатки, а также области применения.
2. Исследование протоколов уровня приложений:
 - Выберите один из протоколов прикладного уровня (например, HTTP, FTP, SMTP). Исследуйте его структуру, основные команды и процесс работы. Подготовьте краткий отчет.
3. Практическое задание:
 - Установите и настройте локальный веб-сервер (например, Apache или Nginx). Опишите процесс его настройки и подключите клиента к серверу через HTTP.
4. Безопасность протоколов:
 - Исследуйте уязвимости, связанные с основными протоколами передачи данных, и подготовьте рекомендации по их защите.

5. Практическое программирование:

- Напишите простую программу на Python, реализующую клиент-серверное взаимодействие по протоколу TCP. Опишите структуру кода и функциональные возможности.

Контрольные вопросы по теме

1. Какова роль протоколов передачи данных в компьютерных сетях?
2. В чем различия между протоколами TCP и UDP?
3. Какие основные протоколы используются для передачи данных на уровне приложений, и каковы их функции?
4. Почему важна маршрутизация IP-пакетов, и как она осуществляется?
5. Каковы основные угрозы безопасности, связанные с протоколами передачи данных, и какие меры можно принять для их предотвращения?

Самостоятельная работа № 28 (2 часа)

Тема: Стеки протокола FTP

Цель работы: Понять протокол FTP и его применение для обмена файлами в сетевой среде.

Краткие сведения из теории

FTP (File Transfer Protocol) — протокол, используемый для передачи файлов по сети TCP/IP. Он позволяет пользователям загружать и скачивать файлы на удалённые серверы, а также управлять файлами на этих серверах. FTP работает в клиент-серверной модели, где клиент отправляет запросы, а сервер обрабатывает эти запросы и пересылает файлы.

Существует несколько уровней протокола FTP:

1. Уровень передачи:
 - Осуществляет соединение и управление сессией. FTP использует две основные транмиссионные модели:
 - Активный режим (порт 21) — клиент инициирует контрольное соединение, а сервер устанавливает соединение для передачи данных.
 - Пассивный режим (также порт 21) — клиент инициирует обе сессии, что особенно полезно, когда клиент находится за файрволом.
2. Уровень управления:
 - Обрабатывает команды и статусные сообщения. Команды FTP, такие как USER, PASS, LIST и RETR, позволяют пользователям взаимодействовать с сервером. Статусные коды (например, 200, 550) информируют клиента о состоянии выполнения команд.
3. Уровень передачи данных:
 - Отвечает за фактическую передачу файлов между клиентом и сервером. Передача может осуществляться в двух режимах: текстовом и бинарном, в зависимости от типа передаваемых данных.

FTP не шифрует данные, что делает его небезопасным для передачи конфиденциальной информации. Для повышения безопасности используются протоколы, такие как FTPS и SFTP, которые добавляют шифрование к стандартному FTP.

Задания для выполнения студентами

1. Сравнительный анализ:
 - Подготовьте сравнительную таблицу FTP, FTPS и SFTP. Укажите ключевые отличия в функциональности и безопасности.
2. Настройка FTP-сервера:
 - Проведите установку и настройку FTP-сервера (например, vsftpd на Linux). Опишите процесс подключения клиента к серверу и демонстрацию загрузки/скачивания файла.
3. Команды FTP:
 - Исследуйте основные команды FTP. Подготовьте справочник с примерами использования команд для различных операций (например, загрузка, скачивание, удаление).
4. Вопросы безопасности:
 - Изучите угрозы, связанные с использованием FTP, и подготовьте список рекомендаций для обеспечения безопасности при использовании протокола.
5. Практическое задание:
 - Создайте проект, в котором реализуете простой клиент FTP на Python. Опишите свой код и основные функции, позволяющие взаимодействовать с FTP-сервером.

Контрольные вопросы по теме

1. В чем основные функции и рабочие процессы протокола FTP?

2. Каковы различия между активным и пассивным режимами FTP?
3. Какие команды FTP являются наиболее распространенными и как они используются?
4. Почему FTP считается небезопасным и какие меры можно предпринять для повышения его безопасности?
5. Каковы основные особенности и преимущества использования SFTP и FTPS по сравнению с традиционным FTP?

Самостоятельная работа № 29 (2 часа)

Тема: Стеки протокола SSH.

Цель работы: Понять работу протокола SSH и его применения для безопасного удаленного доступа.

Краткие сведения из теории

SSH (Secure Shell) — это протокол, обеспечивающий безопасный доступ к удалённым системам и управлению ими. Он шифрует данные, предотвращая их перехват и несанкционированный доступ, и предоставляет различные механизмы аутентификации. Стеки протокола SSH можно представить в виде нескольких слоёв, каждый из которых выполняет свою функцию:

1. **Транспортный уровень:** Обеспечивает шифрование, целостность и аутентификацию. Применяются алгоритмы, такие как AES и Blowfish. Он создает надёжное соединение между клиентом и сервером.
2. **Аутентификация:** На этом уровне осуществляется верификация пользователя. Возможные методы аутентификации включают:
 - Пароли
 - Публичные ключи
 - Паттерны аутентификации (например, две факторы)
3. **Сессия:** Устанавливает параметры для связи между клиентом и сервером после успешной аутентификации. Это включает в себя управление портами, запуск удалённых команд и обмен данными.

SSH также позволяет наладить безопасный туннелирование трафика (портов), что позволяет передавать данные по небезопасным сетям с минимальными рисками.

Задания для выполнения студентами

1. **Исследование протоколов:**
 - Подготовьте сравнительный анализ протоколов SSH и Telnet, выделяя основные различия в области безопасности и функциональности.
2. **Шифрование:**
 - Изучите различные алгоритмы шифрования, используемые в протоколе SSH, и подготовьте отчет о том, как они обеспечивают безопасность данных.
3. **Настройка SSH:**
 - Проведите практическое задание по установке и настройке SSH-сервера на виртуальной машине. Опишите шаги по подключению и настройке ключевой аутентификации.
4. **Методы аутентификации:**
 - Исследуйте и опишите различные методы аутентификации, поддерживаемые SSH. Подготовьте краткий отчет по каждому методу с примерами применения.
5. **Безопасность и уязвимости:**
 - Проанализируйте основные уязвимости протокола SSH и методы защиты от них. Составьте список рекомендаций для повышения безопасности при использовании SSH.

Контрольные вопросы по теме

1. Каковы основные функции протокола SSH?
2. В чем различия между аутентификацией по паролю и аутентификацией по ключу в SSH?
3. Почему использование SSH предпочтительнее по сравнению с Telnet для удалённого управления системами?
4. Какие алгоритмы шифрования наиболее часто используются в SSH?
5. Какие меры можно предпринять для обеспечения безопасности SSH-соединений?

Самостоятельная работа № 30 (2 часа)

Тема: Серверные дистрибутивы операционных систем.

Цель работы: Понять различные серверные дистрибутивы операционных систем и их использование в современных информационных системах

Краткие сведения из теории

Серверные дистрибутивы операционных систем — это специальные версии операционных систем, оптимизированные для работы на серверах. Они обеспечивают надежность, безопасность и высокую производительность, что критично для обработки больших объемов данных и обеспечения доступности ресурсов.

Наиболее распространенные серверные дистрибутивы:

1. Linux:
 - Ubuntu Server: Популярная дистрибутивная версия Linux, известная своей простотой в установке и использовании. Подходит для виртуальных серверов, облачного хостинга и веб-серверов.
 - CentOS: Дистрибутив, ориентированный на стабильность, часто используется в корпоративной среде. На его базе построен Red Hat Enterprise Linux (RHEL), обладая совместимостью и надежными обновлениями.
 - Debian: Обеспечивает высокую стабильность и надежность. Часто используется для сервера, обеспечивающего базовые сетевые услуги.
2. Windows Server:
 - Windows Server 2022: Актуальная на текущий момент (2024 год) версия Windows Server, предлагающая усовершенствования в области безопасности, управления и функциональности для облачных технологий и виртуализации.
3. FreeBSD: Дистрибутив, основанный на UNIX. Известен своей производительностью и надежностью, часто используется для веб-серверов и сетевых приложений.

Серверные дистрибутивы обычно предоставляют функции, такие как управление пользователями, службы безопасности, поддержку виртуализации и совместимость с различными приложениями и базами данных.

Задания для выполнения студентами

1. Сравнительный анализ дистрибутивов:
 - Подготовьте сравнительную таблицу серверных дистрибутивов (например, Ubuntu Server, CentOS, Windows Server и FreeBSD) по таким критериям, как характеристики, область применения, поддержка и обновления.
2. Системные требования:
 - Исследуйте системные требования для установки выбранного вами серверного дистрибутива. Подготовьте краткий отчет о необходимых ресурсах.
3. Установка дистрибутива:
 - Создайте пошаговую инструкцию по установке одного из серверных дистрибутивов (Linux или Windows Server). Опишите процесс конфигурации после установки.
4. Безопасность серверов:
 - Изучите основные рекомендации по безопасности для серверных дистрибутивов. Составьте список лучших практик по защите серверов от угроз.
5. Применение в реальных задачах:
 - Исследуйте примеры использования различных серверных дистрибутивов в реальных проектах (например, веб-хостинг, базы данных, облачные решения) и подготовьте краткий обзор.

Контрольные вопросы по теме

1. В чем ключевые отличия между серверными и десктопными дистрибутивами операционных систем?
2. Какие основные функции предоставляет серверный дистрибутив?
3. Почему CentOS и Debian считаются надежными выборами для серверов?
4. Какие особенности Windows Server 2022 выделяют его среди других серверных дистрибутивов?
5. Какие меры можно предпринять для повышения безопасности серверной операционной системы?

Самостоятельная работа № 31 (2 часа)

Тема: Проводной и беспроводной доступ к сети: устройства и кабели.

Цель работы: Понять принципы работы проводных и беспроводных сетей, а также познакомиться с различными устройствами, используемыми для подключения и передачи данных.

Краткие сведения из теории

Проводной доступ к сети предполагает использование физических проводов для соединения сетевых устройств. Наиболее распространенные типы кабелей:

1. Ethernet-кабели (например, категории CAT5e, CAT6) — используются для передачи данных в проводных локальных сетях. Обеспечивают высокую скорость и надежность соединения.
2. Оптоволоконные кабели — используют свет для передачи данных, что позволяет достигать очень высоких скоростей и значительных расстояний без потери качества сигнала. Разделяются на одномодовые и многомодовые.

Беспроводной доступ к сети осуществляется с помощью радиосигналов. Основные технологии:

1. Wi-Fi — наиболее распространенный стандарт для локальных беспроводных сетей, предлагающий различные скорости и диапазоны подключения.
2. Bluetooth — используется для соединения устройств на коротких расстояниях, например, между мобильными телефонами и наушниками.
3. Мобильный интернет (3G, 4G, 5G) — позволяет подключаться к Интернету с помощью мобильных сетей, предлагая различные скорости и охват.

Основные устройства, использующиеся в проводных и беспроводных сетях, включают:

- Маршрутизаторы — обеспечивают связь между разными сетями и управляют трафиком.
- Коммутаторы — соединяют устройства внутри одной сети.
- Точки доступа (AP) — обеспечивают беспроводное соединение для клиентов.
- Модемы — преобразуют сигналы от провайдеров для подключения к сети.

Задания для выполнения студентами

1. Сравнительный анализ:
 - Подготовьте таблицу, в которой сравните проводной и беспроводной доступ к сети по таким критериям, как скорость, надежность, безопасность и стоимость.
2. Типы кабелей:
 - Исследуйте различные типы Ethernet-кабелей и оптоволоконных кабелей. Опишите их характеристики и случаи использования.
3. Практическое задание:
 - Проведите эксперимент: протестируйте скорость Internet-соединения по проводному и беспроводному доступу на своем устройстве. Запишите результаты и сделайте выводы.
4. Исследование оборудования:
 - Изучите функции и характеристики маршрутизатора. Подготовьте краткий отчет о том, как устанавливать и настраивать маршрутизатор для домашней сети.
5. Безопасность сети:
 - Изучите основные меры безопасности для беспроводных сетей и подготовьте список рекомендаций для защиты Wi-Fi-сети от несанкционированного доступа.

Контрольные вопросы по теме

1. Каковы ключевые преимущества и недостатки проводного и беспроводного подключения к сети?
2. Что такое Ethernet-кабель и какие типы существуют?
3. Как работает оптоволоконная связь и в чем ее преимущества?
4. Что такое точка доступа, и какова её роль в беспроводной сети?
5. Какие меры безопасности необходимы для защиты беспроводных сетей?

Самостоятельная работа № 32 (2 часа)

Тема: Адресация в сети. Провайдеры. Понятие хостинга.

Цель работы: Понять вопросы адресации в сети, работы провайдеров и понятий, связанных с хостингом.

Краткие сведения из теории

Адресация в сети — это процесс, при котором каждому устройству в компьютерной сети назначается уникальный адрес (например, IP-адрес), который позволяет идентифицировать его и маршрутизировать данные. Существует два основных типа IP-адресов:

1. IPv4: 32-битные адреса, представляемые в десятичном формате (например, 192.168.1.1). Они способны адресовать около 4 миллиардов устройств.
2. IPv6: 128-битные адреса, созданные для решения проблемы нехватки адресов в IPv4, и представляемые в шестнадцатеричном формате (например, 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

Провайдеры — это организации, которые обеспечивают доступ к сети Интернет. Существует несколько типов провайдеров:

1. Интернет-провайдеры (ISP): Предоставляют услуги подключения к Интернету.
2. Хостинг-провайдеры: Предоставляют серверное пространство и услуги для размещения веб-сайтов и приложений.
3. Телекоммуникационные компании: Обеспечивают передачу информации через телефонные связи и другие средства.

Хостинг — услуга, которая позволяет размещать веб-сайты и приложения на серверах, доступных через Интернет. Хостинг может быть выделенным, виртуальным, облачным или управляемым, и позволяет бизнесам и индивидуальным пользователям иметь доступ к необходимым ресурсам для размещения и управления своим контентом.

Задания для выполнения студентами

1. Исследование адресации:
 - Ознакомьтесь с понятиями IPv4 и IPv6. Подготовьте сравнительный анализ отчасти, включая преимущества и недостатки каждого типа.
2. Типы провайдеров:
 - Составьте список различных типов провайдеров (ISP, хостинг-провайдеры, телекоммуникационные компании) и кратко опишите их функции и услуги, которые они предлагают.
3. Изучение хостинга:
 - Исследуйте различные виды хостинга (выделенный, виртуальный, облачный) и подготовьте отчет о том, в каких случаях целесообразно использовать каждый из типов.
4. Практическое задание: определение IP-адреса:
 - Используйте команду `ipconfig` (Windows) или `ifconfig` (Linux/Mac) для определения своего IP-адреса и запишите полученную информацию. Сравните его с адресами, доступными на сайте, например, whatismyip.com.
5. Провайдеры в вашем регионе:
 - Исследуйте и составьте список интернет-провайдеров, доступных в вашем регионе. Укажите их предложения и тарифы.

Контрольные вопросы по теме

1. Что такое IP-адрес и какова его роль в компьютерной сети?
2. В чем ключевые отличия между IPv4 и IPv6?
3. Каковы основные функции интернет-провайдеров?
4. Какие существуют виды хостинга и как выбрать подходящий для размещения веб-сайта?
5. Как повлиял переход на IPv6 на развитие Интернета?

8. ЗАКЛЮЧЕНИЕ

Методические рекомендации разработаны в соответствии с требованиями рабочей учебной программы дисциплины «Операционные системы и среды».

Организация самостоятельной работы является ключевым элементом в подготовке высококвалифицированных специалистов в области компьютерных систем. Учебный процесс направлен на развитие у студентов не только теоретических знаний, но и практических навыков.

Студенты обеспечены разнообразными заданиями, которые охватывают различные аспекты работы с операционными системами. Задания включают не только теоретические аспекты, но и практические упражнения, позволяющие студентам применять полученные знания на практике. Создание, настройка и управление виртуальными машинами, выполнение задач по администрированию и управлению файловыми системами. Студенты стимулируются к активному изучению материала, включению в обсуждения и решение проблем, что можно достигнуть через групповую работу, проектные задания и участие в обсуждениях. Это помогает развить критическое мышление и аналитические способности.

Оценка результатов самостоятельной работы является многогранной, включая как формальные, так и неформальные методы. Используются тесты, рефераты, практические задания и проекты, чтобы обеспечить всестороннюю оценку знаний и навыков студентов.

Регулярная обратная связь со стороны преподавателя позволяет студентам корректировать свои действия и препятствовать появлению ошибок в процессе обучения. Обсуждение выполненных заданий помогает углубить понимание материала и выявить зоны для улучшения.

Таким образом, эффективная организация самостоятельной работы помогает обучающимся не только овладеть необходимыми знаниями и навыками, но и подготовит их к профессиональной деятельности в области компьютерных систем и комплексов. Такой подход создаст фундамент для дальнейшего успешного обучения и профессионального роста студентов.

9. ПРИЛОЖЕНИЕ

1. Требования к оформлению реферата

Объем реферата – 20 – 25 стр. печатного текста. Шрифт – не более 14 pt, TimesNewRoman, интервал – 1,5, поля: верхнее, нижнее, левое – 2 см, правое 1,5 см.

На титульном листе указывается название работы, ФИО студента и группа, ФИО преподавателя (научного руководителя), проверяющего и оценивающего реферат, наименование кафедры и учебного заведения. Тема реферата может быть сформулирована самостоятельно, по согласованию с преподавателем.

Название работы оформляется следующим образом:

Реферат по дисциплине «Операционные системы и среды» на тему: «.....»

Текст реферата печатается на одной стороне страницы; сноски и примечания печатаются на той же странице, к которой они относятся (через 1 интервал, более мелким шрифтом, чем текст). Основной текст должен сопровождаться иллюстративным материалом (рисунки, фотографии, диаграммы, схемы, таблицы, программы). Если в основной части содержатся цитаты или ссылки на высказывания, необходимо указать номер источника по списку, приведенному в конце реферата, и страницу в квадратных скобках в конце цитаты или ссылки.

Реферат – это краткое изложение в письменной форме содержания прочитанных книг и документов; сообщение об итогах изучения научного вопроса; доклад на определенную тему, освещающий ее вопросы на основе литературных и других источников. Целью написания реферата является углубление знаний по конкретной проблеме, получение навыков работы с научной и научно-популярной литературой. Работа над рефератом требует, как правило, не менее месяца.

В процессе работы над проблемой необходимо:

- вычленив проблему;
- самостоятельно изучить проблему на основе первоисточников;
- дать обзор использованной литературы;
- последовательно и доказательно изложить материал;
- правильно оформить ссылки на источники.

2. Обязательные структурные элементы реферата:

1. Введение, в котором описывается актуальность проблемы, определяются цели и задача реферата; объем введения – 1 - 2 страницы.
2. Содержание.
3. Текст реферата должен содержать:
 - обоснование выбранной темы;
 - сравнительный анализ литературы по проблеме;
 - изложение собственной точки зрения на проблему;
 - выводы и предложения;
 - заключение.

4. Список использованных источников должен оформляться в соответствии с ГОСТом и может содержать не только названия книг, журналов, газет, но и любые источники информации (например, сведения из сети Интернет, информацию из теле- и радиопередач, а также частные сообщения каких-либо специалистов, высказанные в личных беседах их с автором реферата).

Реферат излагается доступным научным (научно-популярным) языком в относительно сжатой форме с использованием облегченных синтаксических конструкций. Такие конструкции могут стать своеобразным планом реферативной статьи: “ В рассматриваемой статье ставится ряд вопросов ...Автор подчеркивает, что ... Более подробно рассмотрена проблема... Анализируются разные точки зрения ... В заключение необходимо отметить что ...” и т.д.

При выставлении оценки за реферат учитываются следующие компоненты:

- содержательная часть (глубина проработки проблемы, структура работы, объем проанализированных источников и т.п.);
- оформление (соответствие стандарту, эстетика оформления, наличие иллюстративного материала и т.п.);
- защита реферата (ориентация в тексте реферата, ответы на вопросы и т.п.).

Реферат сдается в отпечатанном виде и на электронном носителе.

3. Темы рефератов

1. История развития операционных систем: от первых систем до современных решений.
2. Сравнительный анализ операционных систем: Windows, Linux и macOS.
3. Управление памятью в операционных системах: подходы и технологии.
4. Файловые системы: виды и их особенности.
5. Процесс и управление процессами в операционных системах: состояние, планирование и взаимодействие.
6. Системы виртуализации: принципы и их применение.
7. Безопасность в операционных системах: угрозы и методы защиты.
8. Операционные системы реального времени: принципы работы и области применения.
9. Параллельные и распределенные операционные системы: особенности и примеры.
10. Графические пользовательские интерфейсы (GUI) в операционных системах: принципы проектирования и разработки.
11. Сетевые операционные системы и их роль в современных вычислительных средах.
12. Энергетические менеджеры в мобильных операционных системах: от Android до iOS.
13. Микроядерные и монолитные архитектуры операционных систем: преимущества и недостатки.
14. Системы на основе облачных технологий: принципы работы и примеры.
15. Контейнеризация и оркестрация в операционных системах: технологии Docker и Kubernetes.

10. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Основные источники:

1. Кудрина, Е. В. Операционные системы и среды на языке С# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2024. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/541725> (дата обращения: 04.09.2024).

2. Трофимов, В. В. Операционные системы и среды : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2018. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/422888> (дата обращения: 04.09.2024).

3. Борисенко В.В. Основы программирования [Электронный ресурс] / В.В. Борисенко. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 323 с. — 978-5-9556-00039-0. — Режим доступа: <http://www.iprbookshop.ru/52206.html>

4. Мейер Б. Инструменты, алгоритмы и структуры данных [Электронный ресурс] / Б. Мейер. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 542 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73680.html>

Интернет-ресурсы

www.ttgt.org (Сайт Тихорецкого Техникума Железнодорожного Транспорта)

www.studentlibrary.ru (Электронная библиотека)

www.urait.ru (Электронная библиотека)

www.fcior.edu.ru (Федеральный центр информационно-образовательных ресурсов — ФЦИОР).

www.school-collection.edu.ru (Единая коллекция цифровых образовательных ресурсов).

www.intuit.ru/studies/courses (Открытые интернет-курсы «Интуит» по курсу «Информатика»).

www.ru.iite.unesco.org/publications (Открытая электронная библиотека «ИИТО ЮНЕСКО» по ИКТ в образовании).